

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

І. С. ТВОРОШЕНКО

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни

ОСНОВИ МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ

*(для студентів 2 курсу заочної форми навчання напряму підготовки
6.080101 – Геодезія, картографія та землеустрій)*

Харків
ХНУМГ ім. О. М. Бекетова
2015

Творошенко І. С. Конспект лекцій з дисципліни «Основи моделювання складних систем» (для студентів 2 курсу заочної форми навчання напрямку підготовки 6.080101 – Геодезія, картографія та землеустрій) / І. С. Творошенко; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2015. – 67 с.

Автор канд. техн. наук, доцент І. С. Творошенко

Рецензент д-р. техн. наук, професор К. О. Метешкін

Рекомендовано кафедрою геоінформаційних систем, оцінки землі та нерухомого майна, протокол № 1 від 29 серпня 2015 року.

© І. С. Творошенко, 2015

© ХНУМГ ім. О. М. Бекетова, 2015

ЗМІСТ

Вступ.....	4
1 Класифікація та принципи побудови моделей.....	6
1.1 Основні поняття та принципи побудови математичних та імітаційних моделей.....	6
1.2 Особливості застосування інструментальних засобів для побудови моделей геоінформаційних систем.....	18
2 Імітаційне моделювання.....	28
2.1 Моделювання паралельних процесів.....	28
2.2 Основні поняття та принципи побудови моделей, заснованих на знаннях.....	37
2.3 Застосування інструментальних програмних засобів для моделювання об'єктів та процесів геоінформаційних систем.....	51
Перелік використаних джерел.....	66

ВСТУП

Моделювання є найбільш ефективним способом дослідження складних систем різного призначення – технічних, економічних, екологічних, соціальних, інформаційних як на етапі їх проектування, так і в процесі експлуатації. Можливості моделювання систем не вичерпані, тому постійно з'являються найновіші методи та технології моделювання.

Моделювання як спосіб пізнання використовувалось людиною з давніх часів. Але з появою комп'ютера моделювання систем збагатилось появою принципово нових методів моделювання таких, як аналітичне та імітаційне моделювання. Моделі і методи моделювання використовуються під час створення систем автоматизованого проектування, систем прийняття рішень, систем автоматизованого керування, систем штучного інтелекту [1].

Необхідність у розв'язанні задач моделювання систем виникає не тільки у науковця, але і у проектувальника, виробника, ділової людини під час повсякденної праці.

Сучасні технології моделювання не тільки полегшили і прискорили процес побудови та дослідження моделі, але і значно наблизили сприйняття інформації. Результати моделювання, які представлені засобами тривимірної анімації, допомагають знайти спільну мову і розуміння між спеціалістами з моделювання систем та спеціалістами, що працюють у галузі, яка моделюється.

Дисципліну «Основи моделювання складних систем» студенти вивчають на 2 курсі денної і заочної форм навчання напряму підготовки 6.080101 – Геодезія, картографія та землеустрій.

Метою викладання навчальної дисципліни «Основи моделювання складних систем» є формування знань про принципи побудови математичного забезпечення та аналізу алгоритмів функціонування інтелектуальних динамічних систем, а також навичок застосування інструментальних програмних засобів для моделювання об'єктів та процесів.

Завданням вивчення дисципліни «Основи моделювання складних систем» є набуття практичних навичок побудови моделей геоінформаційних систем за різними методологіями.

Для засвоєння основ моделювання складних систем потрібно володіти знаннями з вищої математики, основ теорії систем, інформатики, програмування. Досконале володіння універсальною мовою програмування та знання методів об'єктно-орієнтованого програмування необхідне для того, щоб не тільки створювати моделі систем, але і створювати системи моделювання.

У результаті вивчення навчальної дисципліни студент повинен мати компетентності пов'язані із:

- здатністю демонструвати поглиблені знання в галузі математики і природничих наук;

- здатністю використовувати вільне володіння професійно-профільованими знаннями в галузі інформаційних технологій, сучасних комп'ютерних мереж, програмних продуктів і ресурсів інтернет для вирішення завдань професійної діяльності, у тому числі, що знаходяться за межами профільної підготовки;

- здатністю самостійно здобувати за допомогою інформаційних технологій і практичної діяльності нові знання та вміння, у тому числі в нових галузях знань, безпосередньо не пов'язаних зі сферою діяльності, розширювати і поглиблювати свій науковий світогляд;

- здатністю вільно володіти професійними знаннями для аналізу і синтезу фізичної інформації (відповідно до профілю підготовки).

Для досягнення мети вивчення дисципліни «Основи моделювання складних систем» необхідно сформувати у майбутніх спеціалістів професійні навички та вміння з практичного застосування методів синтезу та аналізу імітаційних моделей систем різного рівня складності.

У процесі вивчення дисципліни студенти повинні продемонструвати вміння будувати імітаційні моделі у залежності від їх структури та функціонування, досліджувати поведінку системи на моделі у залежності від умов протікання процесів у системі і при різних значеннях параметрів, застосовуючи для цього існуючі теоретичні методи та інструментальні засоби.

Однією із основних вимог до імітаційної моделі є її адекватність та максимальна наближеність алгоритмів до методології об'єктно-орієнтованого програмування, а також універсальність, як можливість подання всієї множини дискретних станів системи.

У конспекті лекцій з дисципліни «Основи моделювання складних систем» розглянуто основні питання застосування мереж Петрі для побудови та аналізу імітаційних моделей систем, наведено загальну характеристику мереж Петрі, правила їх маркірування та виконання вводу, класифікацію та розширення мереж, методи аналізу мережевих моделей, приклади побудови моделей для типових задач.

1 КЛАСИФІКАЦІЯ ТА ПРИНЦИПИ ПОБУДОВИ МОДЕЛЕЙ

1.1 Основні поняття та принципи побудови математичних та імітаційних моделей

План

1. Основні поняття та означення.
2. Основні поняття та принципи побудови математичних моделей.
3. Основні поняття та принципи побудови імітаційних моделей.

У даний час не можна назвати галузь людської діяльності, в якій у тій чи іншій мірі не використовувалися б методи моделювання. Особливо це стосується сфери управління різними системами, де основними є процеси прийняття рішень на основі отриманої інформації.

Гіпотези та аналогії, що відображають реальний, об'єктивно існуючий світ, повинні володіти наочністю або зводитися до зручних для дослідження логічних схем, що спрощують міркування, логічні побудови та дозволяють проводити експерименти, уточнюючи природу явищ, називаються *моделями* [2].

У загальному випадку модель має структуру, зображену на рисунку 1.1.

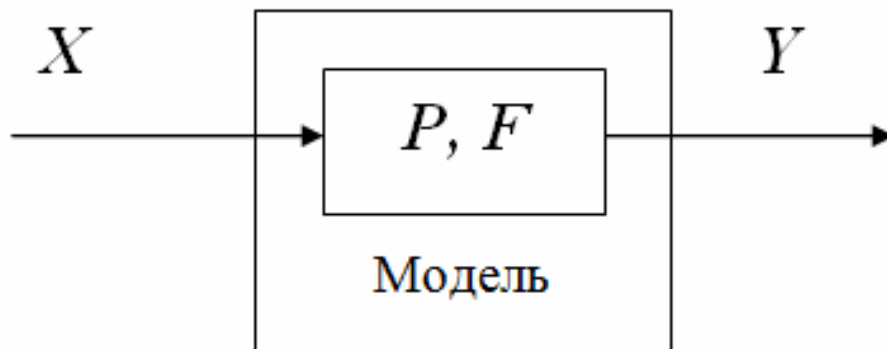


Рисунок 1.1 – Загальна структура моделі

На рисунку 1.1 введено такі позначення:

- X – множина входних змінних системи;
- Y – множина вихідних змінних системи;
- P – множина параметрів;
- F – функція, функціонал, алгоритм або формальне подання залежності змінних Y від змінних X .

Модель – це об’єкт-заступник об’єкта-оригіналу, що забезпечує вивчення деяких властивостей оригіналу. Заміщення одного об’єкта іншим для одержання інформації про найважливіші властивості об’єкта-оригіналу за допомогою об’єкта-моделі називається *моделюванням* [2].

Таким чином, моделювання може бути визначене як подання об’єкта моделлю для отримання інформації про цей об’єкт шляхом проведення експериментів з його моделлю. Теорія заміщення одних об’єктів (оригіналів) іншими об’єктами (моделями) і дослідження властивостей об’єктів на їхніх моделях називається *теорією моделювання* [1].

Існують два способи побудови моделей. При першому способі в результаті ретельного вивчення системи встановлюються закони функціонування системи, які потім відтворюються за допомогою моделі. Поведінку системи, таким чином, досліджують на моделі. Параметри моделі P у цьому випадку пов’язані з реальними процесами і мають фізичну інтерпретацію. Моделі такого типу називають *фізичними* моделями [2].

При другому способі припускається вид залежності F , невідомі параметри якої P потім відшукуються за даними спостережень за змінними системи X , Y . Параметри P в цьому випадку не зв’язані з фізикою реальних процесів, що протікають в системі, цей зв’язок досліднику залишається невідомим. Тому моделі такого типу називають *нефізичними* моделями.

Розглянемо види моделювання систем [2].

За рівнем повноти моделі діляться на *повні*, *неповні* і *наближені* (рис. 1.2). Повні моделі ідентичні об’єкту в часі і просторі, для неповного моделювання ця ідентичність не зберігається. Під час наближеного моделювання деякі сторони функціонування реального об’єкта не моделюються зовсім.

У залежності від характеру досліджуваних процесів у системі моделювання буває *детерміноване* та *стохастичне*, *статистичне* та *динамічне*, *дискретне*, *дискретно-безперервне* та *безперервне*.

Детерміноване моделювання відображає процеси, в яких передбачається відсутність випадкових впливів. Стохастичне моделювання враховує ймовірнісні процеси та події. Статистичне моделювання служить для опису поведінки об’єкта у фіксований момент часу, а динамічне – для дослідження об’єкта у часі.

Дискретне, дискретно-безперервне та безперервне моделювання використовується для опису процесів, що мають зміну в часі та оперують аналоговими, аналогово-цифровими і цифровими моделями.

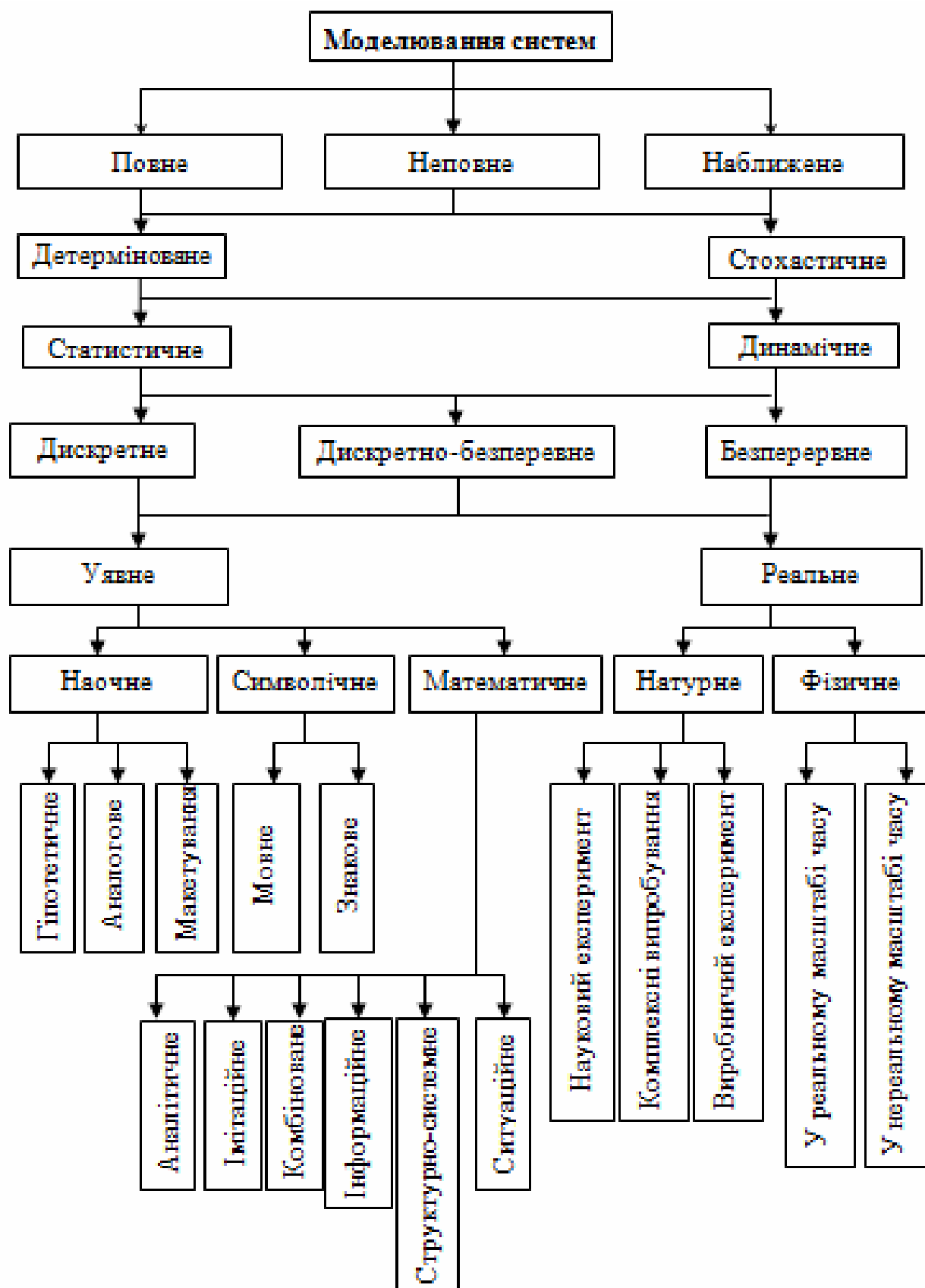


Рисунок 1.2 – Види моделювання систем

У залежності від форми подання об'єкта моделювання класифікується на *уявне* і *реальне*. Уявне моделювання застосовується тоді, коли моделі не реалізуються у заданому інтервалі часу або відсутні умови для їх фізичного створення.

Уявне моделювання реалізується у вигляді наочного, символічного і математичного.

При *наочному* моделюванні на базі уявлень людини про реальні об'єкти створюються наочні моделі, які відображатимуть явища і процеси, що протікають в об'єкті.

В основу *гіпотетичного* моделювання закладається гіпотеза про закономірності протікання процесу в реальному об'єкті, яка відображає рівень знань дослідника про об'єкт і базується на причинно-наслідкових зв'язках між входом і виходом досліджуваного об'єкта. Цей вид моделювання використовується, коли знань про об'єкт достатньо для побудови формальних моделей.

Аналогове моделювання ґрунтується на застосуванні аналогій різних рівнів. Для простих об'єктів найвищим рівнем є повна аналогія, з ускладненням системи використовуються аналогії таких рівнів, коли аналогова модель відображає кілька або тільки одну сторону функціонування об'єкта.

Макетування застосовується, коли в реальному об'єкті протікають процеси, що не піддаються фізичному моделюванню або можуть передувати проведенню інших видів моделювання.

Символічне моделювання – це штучний процес створення логічного об'єкта, який заміщає реальний і виражає основні властивості його відношень за допомогою певної системи знаків і символів.

В основі *мовного* моделювання лежить деякий тезаурус, який утворюється з набору вхідних понять, причому цей набір повинен бути фіксованим. Між тезаурусом і звичайним словником є принципові відмінності. *Тезаурус* – словник, який очищено від неоднозначності, тобто у ньому кожному слову може відповідати лише єдине значення, хоча в звичайному словнику одному слову може відповідати кілька значень.

Якщо ввести умовне позначення окремих понять, тобто знаки, а також певні операції між цими знаками, то можна реалізувати *знакове* моделювання і за допомогою знаків відображати набір понять – скласти окремі ланцюжки із слів і пропозицій. Використовуючи операції об'єднання, перетину і доповнення теорії множин, можна в окремих символах дати опис якогось реального об'єкта.

Математичне моделювання – це процес встановлення відповідності даному реальному об'єкту деякого математичного об'єкта [3].

Для дослідження характеристик процесу функціонування будь-якої системи математичними методами повинна бути обов'язково проведена формалізація цього процесу, тобто побудована математична модель. Будь-яка математична модель описує реальний об'єкт з деякою мірою наближення.

Для *аналітичного* моделювання характерно те, що процеси функціонування елементів системи записуються у вигляді деяких функціональних співвідношень або логічних умов.

Аналітична модель досліджується такими методами:

- аналітичним, коли прагнуть отримати в загальному вигляді явні залежності, що зв'язують шукані характеристики з початковими умовами, параметрами і змінними системи;

- чисельним, коли, не вміючи вирішувати рівняння у загальному вигляді, прагнуть отримати числові результати при конкретних початкових даних;

- якісним, коли, не маючи рішення в явному вигляді, можна знайти деякі властивості рішення (наприклад, оцінити стійкість рішення).

У даний час поширені методи машинної реалізації дослідження характеристик процесу функціонування складної системи.

При *імітаційному* моделюванні алгоритм, що реалізує модель, відтворює процес функціонування системи у часі, причому імітуються елементарні явища, що становлять процес, зі збереженням їх логічної структури і послідовності протікання у часі, що дозволяє за вихідними даними отримати відомості про стани процесу в певні моменти часу, що дають можливість оцінити характеристики системи.

Основною перевагою імітаційного моделювання у порівнянні з аналітичним є можливість вирішення більш складних завдань. Імітаційні моделі дозволяють досить просто враховувати такі фактори, як наявність дискретних і безперервних елементів, нелінійні характеристики елементів системи, численні випадкові впливи, які часто створюють труднощі під час аналітичних досліджень. У даний час імітаційне моделювання – найбільш ефективний метод дослідження складних систем та єдиним доступним методом отримання інформації про поведінку системи, особливо на етапі її проектування.

В імітаційному моделюванні розрізняють метод статистичного моделювання і метод статистичних випробувань (Монте-Карло). Якщо результати, отримані під час відтворення на імітаційній моделі, є реалізаціями випадкових величин і функцій, тоді для знаходження характеристик процесу потрібно його багаторазове відтворення з подальшою обробкою інформації. Тому доцільно використовувати метод статистичного моделювання.

Спочатку було розроблено метод статистичних випробувань, що є чисельним методом, який застосовувався для моделювання випадкових величин і функцій, ймовірнісні характеристики яких збігалися з рішеннями аналітичних завдань (така процедура отримала назву – метод Монте-Карло). Потім цей прийом стали застосовувати і для машинної імітації з метою дослідження характеристик процесів функціонування систем, схильних до випадкових впливів, тобто з'явився метод статистичного моделювання.

Метод імітаційного моделювання застосовується для оцінки варіантів структури системи, ефективності різних алгоритмів управління системою, впливу зміни різних параметрів системи. Імітаційне моделювання може бути покладено в основу структурного, алгоритмічного та параметричного синтезу складних систем, коли потрібно створити систему з заданими характеристиками. Така система повинна бути оптимальною за деякими критеріями ефективності.

Комбіноване (аналітико-імітаційне) моделювання дозволяє об'єднати аналітичне та імітаційне моделювання. Під час побудови комбінованих моделей проводиться попередня декомпозиція процесу функціонування об'єкта на складові підпроцеси і для тих з них, де це можливо, використовуються аналітичні моделі, а для решти підпроцесів будуються імітаційні моделі. Такий підхід дозволяє охопити якісно нові класи систем, які не можуть бути досліджені з використанням тільки аналітичного або імітаційного моделювання окремо.

Інформаційне моделювання (кібернетичне) пов'язане з дослідженням моделей, в яких відсутня безпосередня подібність фізичних процесів, що відбуваються у моделях, реальним процесам. У цьому випадку прагнуть відобразити лише деяку функцію і розглядають реальний об'єкт як «чорний ящик», що має ряд входів і виходів, та моделюються деякі зв'язки між виходами і входами. Таким чином, в основі інформаційних (кібернетичних) моделей лежить відбиток деяких інформаційних процесів управління, що дозволяє оцінити поведінку реального об'єкта. У цьому випадку для побудови моделі необхідно виділити досліджувану функцію реального об'єкта, спробувати формалізувати цю функцію у вигляді деяких операторів зв'язку між входом та виходом, відтворити дану функцію на імітаційній моделі, причому на абсолютно іншій математичній мові.

Структурно-системне моделювання базується на деяких специфічних особливостях структур певного виду, використовуючи їх як засіб дослідження систем або розробляючи на їх основі, із застосуванням інших методів формалізованого подання систем, специфічні підходи до моделювання.

Структурно-системне моделювання включає [4]:

- методи мережевого моделювання;
- поєднання методів структуризації з лінгвістичними (мовними);
- структурний підхід у напрямку формалізації побудови та дослідження структур різного типу (ієрархічних, матричних, графів) на основі теоретико-множинних подавань та поняття номінальної шкали теорії вимірювань.

Ситуаційне моделювання ґрунтується на модельній теорії мислення, у рамках якої можна описати основні механізми регулювання процесів прийняття рішень. В основі модельної теорії мислення лежить уявлення про формування у структурах мозку інформаційної моделі об'єкта і зовнішнього світу. Ця інформація сприймається людиною на базі вже наявних у неї знань і досвіду. Поведінка людини будується шляхом формування цільової ситуації і уявного перетворення вихідної ситуації в цільову. Основою побудови моделі є опис об'єкта у вигляді сукупності елементів, пов'язаних між собою певними відношеннями, що відображають семантику предметної області.

При *реальному* моделюванні використовується можливість дослідження характеристик або на всьому реальному об'єкті, або на його частині. Такі дослідження проводяться як на об'єктах, що працюють в нормальних режимах, так і при організації спеціальних режимів для оцінки характеристик, які цікавлять дослідника. Реальне моделювання є найбільш адекватним, але його можливості обмежені.

Натурним моделюванням називають проведення дослідження на реальному об'єкті з наступною обробкою результатів експерименту на основі теорії подібності. Натурний експеримент підрозділяється на науковий експеримент, комплексні випробування і виробничий експеримент.

Науковий експеримент характеризується широким використанням засобів автоматизації проведення, застосуванням досить різноманітних засобів обробки інформації, можливістю втручання людини в процес проведення експерименту.

Один із різновидів експерименту – *комплексні випробування*, коли внаслідок повторення випробувань об'єктів в цілому (або великих частин системи) виявляються загальні закономірності про характеристики якості та надійності цих об'єктів. У цьому випадку моделювання здійснюється шляхом обробки та узагальнення відомостей про групу однорідних явищ.

Поряд зі спеціально організованими випробуваннями можлива реалізація натурального моделювання шляхом узагальнення досвіду, накопиченого у ході виробничого процесу, тобто можна говорити про *виробничий експеримент*.

На базі теорії подібності обробляють статистичний матеріал по виробничому процесу і отримують його узагальнені характеристики. Необхідно пам'ятати про відмінність експерименту від протікання реального процесу. Вона полягає в тому, що в експерименті можуть з'явитися окремі критичні ситуації та визначитись кордони стійкості процесу. У ході експерименту вводяться нові фактори і впливи в процес функціонування об'єкта.

Іншим видом реального моделювання є *фізичне*, яке відрізняється від натурного тим, що дослідження проводиться на установках, які зберігають природу явищ і володіють фізичною схожістю. У процесі фізичного моделювання задаються деякі характеристики зовнішнього середовища і досліджується поведінка реального об'єкта, або його моделі при заданих, або створених штучно, впливах зовнішнього середовища.

Фізичне моделювання може протікати у *реальному* та *нереальному* масштабах часу або розглядатися без урахування часу. В останньому випадку вивченню підлягають так звані «заморожені» процеси, що фіксуються у деякий момент часу. Найбільшу складність та зацікавленість з точки зору коректності одержаних результатів надає фізичне моделювання в реальному масштабі часу.

Однакові об'єкти моделювання у залежності від мети та задачі дослідження можуть мати різні моделі.

Серед задач моделювання виділимо такі задачі [4]:

- задача моделювання;
- задача управління;
- задача ідентифікації;
- задача оптимізації;
- задача прогнозування.

Задача моделювання (пряма задача) полягає у знаходженні значень вихідних змінних Y при відомих значеннях вхідних змінних X , відомій моделі F та визначених параметрах P (рис. 1.1).

Задача управління (зворотна задача) полягає у знаходженні таких значень вхідних змінних X , що забезпечують задані значення вихідних змінних Y при відомій моделі F та фіксованих значеннях параметрів P .

У формулюванні *задачі ідентифікації* відомими є множина вхідних змінних X , множина вихідних змінних Y та множина моделей F . Потрібно визначити єдину модель f з множини запропонованих моделей F , і визначити її параметри P , що забезпечують при вхідних значеннях X вихідні значення Y .

У постановці задачі оптимізації відомими є модель F , множина можливих вхідних значень X та критерій оптимізації K , а від дослідника вимагається знайти значення вхідних змінних X , значення параметрів P , та значення вихідних змінних Y , що задовольняють заданому критерію оптимізації K .

Задача прогнозування формулюється так, що при відомих для дослідника значеннях вхідних та вихідних значень моделі X_t , Y_t до моменту часу t та заданому часі прогнозування T потрібно визначити модель F та її параметри P , які забезпечують найкращий прогноз Y_{t+T} .

Серед великої кількості методів моделювання, що існують, виділимо математичне моделювання.

Моделювання аналітичне, якщо подання залежності F вихідних змінних Y від вхідних її змінних X має аналітичний вигляд, тобто у вигляді відомих аналітичних функцій. Функція називається аналітичною, якщо вона розкладається у ряд Тейлора. Аналітичні функції диференційовані безліч разів, тому до них можуть застосовуватись методи математичного аналізу. Перевагою цього методу моделювання є можливість отримати залежність $Y = f(X)$ в явному вигляді і застосувати до неї методи класичного математичного аналізу. Якщо є можливість побудувати аналітичну модель системи, то завжди віддають перевагу цьому методу моделювання.

Знаходження залежності $Y = f(X)$ може виявитись настільки складним, що досліднику доведеться застосувати спеціальне програмне забезпечення, а для деяких систем доведеться відмовитися від пошуку абстрактної залежності $Y = f(X)$ і задовольнитися наближеним розв'язком, що знаходиться чисельними методами. Деякі системи настільки складні, що не дивлячись на те, що опис їх функціонування піддається опису аналітичними функціями, знаходження залежності $Y = f(X)$ у явному вигляді виявляється неможливим. Усі задачі математичного програмування мають досить простий аналітичний опис, але розв'язок задачі може бути знайдений тільки у результаті виконання певної кількості кроків, такий метод моделювання називають *математичним моделюванням*. Алгоритм знаходження точного розв'язку задачі може бути реалізований дослідником самостійно, за допомогою спеціального програмного забезпечення або за допомогою чисельних методів.

Існують системи, опис яких не піддається опису аналітичними функціями, але процес функціонування їх може бути описаний алгоритмом імітації. Під імітацією розуміють відтворення за допомогою комп'ютерної програми процесу функціонування складної системи у часі.

У результаті багатократних прогонів імітаційної моделі дослідник отримує інформацію про властивості реальної системи, такий метод моделювання називають *імітаційним моделюванням*.

Процес моделювання складається з кількох етапів.

На першому етапі дослідник визначає мету та задачу моделювання.

На другому етапі, виходячи з мети та задачі моделювання, дослідник приступає до вербального опису системи. Опис набору змінних моделі, разом із описом структури системи та формулюванням мети та задачі дослідження складає концептуальну модель системи. Виходячи із концептуальної моделі системи та з огляду на вибір інструментальних засобів, дослідник робить вибір теоретичної бази, на основі якої буде побудована модель системи. Отже, обравши теоретичну базу моделювання, дослідник має описати систему, що розглядається, обраними елементами формального опису і визначити для них усі необхідні параметри. Формальне подання системи має вигляд схеми, в якій вказані зв'язки між елементами системи та зв'язки із зовнішнім середовищем, а також вказані параметри елементів системи. У формальній моделі міститься також інформація, яким чином будуть знайдені вихідні змінні моделі у результаті моделювання.

На третьому етапі дослідник приступає до створення моделі. Спочатку виконується реалізація моделі за допомогою обраного програмного забезпечення. Потім виконується верифікація моделі, тобто перевірка алгоритму моделювання на відповідність задуму моделювання. Наприклад, змінюють значення вхідних змінних і спостерігають як модель реагує на таку зміну. Якщо реакція моделі відповідає логіці її функціонування, то модель вважається правильною. Завершується створення моделі перевіркою адекватності моделі, що полягає у порівнянні значень вихідних змінних об'єкта, що моделюється, і моделі при однакових значеннях вхідних змінних. Очевидно, що таку перевірку можна здійснити тільки, якщо відомі деякі значення вхідних і вихідних змінних досліджуваного об'єкта.

Четвертий етап – це дослідження моделі. Результати моделювання стають корисними, якщо проведено змістовне дослідження моделі відповідно до мети моделювання.

Експерименти, що проводяться з моделлю, мають бути спочатку сплановані, потім проведені та статистично оброблені.

Аналіз результатів моделювання складається з оцінки точності результатів моделювання, оцінки стійкості результатів моделювання та оцінки чутливості результатів моделювання.

Формування висновків та пропозицій є завершальним етапом моделювання, на якому підводяться підсумки та висловлюються думки щодо напрямків подальшого дослідження об'єкта моделювання.

Процес моделювання може бути поданим поетапно тільки у звіті про результати моделювання. Поступового наближаючись до мети, досліднику доводиться неодноразово повертатись до попередніх етапів і уточнювати постановку задачі, формальний опис моделі, алгоритм реалізації або план проведення експериментів.

Задачею системного аналізу є формування опису системи, що відповідає меті дослідження системи.

Опис системи складається з опису [5]:

- набору вхідних змінних системи з уточненням їх характеристик;
- набору вихідних змінних системи;
- границь системи з уточненням що є для системи її зовнішнім середовищем;
- елементів системи з уточненням їх основних властивостей;
- зв'язків між елементами системи.

Системний підхід до дослідження систем означає, що дослідник вивчає функціонування системи в цілому, не концентруючи свою увагу на окремих її частинах. Системний підхід ґрунтується на визнанні факту, що функціонування окремих підсистем та елементів системи не гарантує функціонування всієї системи в цілому, оскільки завжди існує взаємодія між частинами системи.

Усім системам притаманні характеристики, що є перепорою на шляху до покращення функціонування системи [6]:

- змінюваність – жодна реальна система не є статичною тривалий проміжку часу, тому модель, що будується, має обмежений термін використання;
- наявність оточуючого середовища – у моделі має бути передбачений вплив зовнішнього середовища, який часто має випадковий характер;
- протиінтуїтивна поведінка – виникає як результат того, що інколи наслідок проявляється пізніше причини;
- тенденція до погіршення характеристик функціонування – окремі частини системи зношуються, у результаті чого погіршується їх робота, що призводить до непередбачуваних наслідків;
- взаємозалежність – усі частини системи залежать одна від одної, тому погіршення характеристик функціонування однієї частини системи впливає на характеристики функціонування інших частин системи;
- організація – існує ієрархія підсистем, що підпорядковується цільовому призначенню системи.

Будь-яка система об'єктивна і, в той же час, суб'єктивна з точки зору вибору границь системи та її елементів. Одному і тому ж самому об'єктивному процесу дослідник може поставити у відповідність різні системи і тільки досвід дослідника, його інтуїція і здатність творчо мислити допомагає здійснити вибір між багатьма варіантами і виконати дослідження системи найкращим способом.

Моделі, побудовані із застосуванням системного підходу, отримали назву *системних моделей* [6].

Опис системи, разом із уточненням мети та задачі дослідження, складає сутність *концептуальної моделі* системи. Назва «концептуальна» походить від латинського слова *conceptio*, що означає «сприйняття».

Виділимо такі етапи створення концептуальної моделі системи [4]:

- визначення мети дослідження системи (орієнтація);
- вибір рівня деталізації системи (стратифікація);
- визначення елементів системи (деталізація);
- визначення впливу зовнішнього середовища (локалізація);
- визначення зв'язків між елементами системи та із зовнішнім середовищем (структуризація).

Моделі можуть бути якісно різними, вони утворюють ієрархію, у якій модель більш високого рівня містить моделі нижчих рівнів як свої частини.

У сучасному світі розв'язання складних наукових та виробничих задач неможливе без використання моделей та моделювання. Серед різних видів моделей особливе місце займають математичні моделі, тому що вони дозволяють враховувати кількісні та просторові параметри явищ та використовувати точні математичні методи.

Вивчення реальних явищ за допомогою математичних моделей, як правило, вимагає застосування обчислювальних методів. Зазвичай, широко використовуються методи прикладної математики, математичної статистики та інформатики.

Для того, щоб вибрати адекватний математичний апарат, призначений для моделювання деякого класу систем, необхідно встановити коло питань, що повинні вирішуватися за допомогою моделей.

Застосування автоматизації проектування за останні роки набуло великого значення у розв'язанні задач різного типу. Невід'ємною частиною систем автоматизованого проектування є математичне забезпечення, яке умовно можна поділити на дві складові частини [1]:

- розрахункові схеми (розрахунки параметрів систем);
- задачі математичного моделювання (дослідження поведінки систем).

1.2 Особливості застосування інструментальних засобів для побудови моделей геоінформаційних систем

План

1. Алгоритмічна модель.
2. Принципи завдання динаміки системи.
3. Моделюючий алгоритм.
4. Основні поняття та принципи побудови моделей геоінформаційних систем.
5. Особливості застосування інструментальних програмних засобів для моделювання об'єктів та процесів геоінформаційних систем.

З давніх часів людина використовує моделювання для дослідження об'єктів, процесів та явищ у різних галузях своєї діяльності. Результати цих досліджень допомагають визначити та покращити характеристики реальних об'єктів та процесів, краще зрозуміти сутність явищ, пристосуватись до них або управляти ними, конструювати нові та модернізувати старі об'єкти. Моделювання допомагає людині приймати обґрунтовані рішення та передбачати наслідки своєї діяльності.

За способом подання моделі можуть бути матеріальні та інформаційні.

Матеріальні моделі – це предметне відображення об'єкта зі збереженням геометричних та фізичних властивостей [6]. Наприклад, географічні карти – це матеріальні моделі реально існуючих об'єктів, такі моделі реалізують матеріальний підхід до вивчення об'єкта чи явища.

Інформаційна модель – це сукупність інформації, що характеризує властивості та стан об'єкта, процесу, явища, а також взаємодію із зовнішнім світом.

За формою подання можна виділити такі види інформаційних моделей:

- геометричні – графічні форми та об'ємні конструкції;
- словесні – усні та письмові описи з використанням ілюстрацій;
- математичні – формули, що показують зв'язок параметрів об'єкта;
- структурні – схеми, графіки, таблиці;
- логічні – моделі, в яких представлені різні варіанти вибору дій на основі різних умов виводу та аналізу;
- спеціальні – хімічні формули;
- комп'ютерні;
- некомп'ютерні.

Поняття комп'ютерного моделювання відображає використання у цьому процесі комп'ютера, як потужного сучасного засобу обробки інформації.

Завдяки комп'ютеру суттєво розширюються галузі застосування моделювання, а також забезпечується всебічний аналіз отриманих результатів.

Поняття, властивості та способи опису алгоритму.

Кожна людина щодня зустрічається з великою кількістю задач, для розв'язання деяких із них існують визначені правила (інструкції, команди). Чим точніше описані правила, тим швидше людина опанує ними і буде ефективніше їх застосовувати, склавши опис деякої послідовності дій для досягнення бажаного результату, тому поняття алгоритму не є новим і незвичайним.

Алгоритмом називається послідовність дій, спрямованих на досягнення зазначеної мети чи на розв'язання поставленої задачі [5].

Будь-який алгоритм повинен мати такі основні *властивості* [5, 6]:

- **детермінованість (визначеність)** – через повну однозначність правил, встановлених в алгоритмі, застосування алгоритму до однакових вхідних даних повинно призводити до однакового результату;

- **дискретність** – процес, що визначається алгоритмом, можна розчленувати (розділити) на окремі елементарні етапи (кроки), кожен з яких називається кроком алгоритмічного процесу чи алгоритму;

- **масовість** – алгоритм повинен бути придатним для розв'язування всіх задач певного типу. Наприклад, алгоритм для розв'язання системи лінійних рівнянь повинен бути придатним для системи, що складається з довільної кількості рівнянь, причому для нього існує множина даних, що допускаються в якості вхідних, тобто початкова система величин може вибиратись із деякої потенційно нескінченної множини;

- **результативність** – вказує на наявність таких варіантів вхідних даних, для яких обчислювальний процес, що реалізується за наданим алгоритмом, повинен через скінчену кількість етапів (кроків) зупинитись і дати шуканий результат або сигнал про те, що наданий алгоритм непридатний для розв'язання поставленої задачі.

Розробка алгоритму складної задачі вимагає високої кваліфікації виконавця і розуміння змісту задачі. З реалізацією алгоритму безпосередньо пов'язане вміння застосувати цей алгоритм до конкретних вхідних даних розв'язуваної задачі. Таке застосування називається *алгоритмічним процесом*. Цей процес полягає у перетворенні вхідних даних за правилами, визначеними заданим алгоритмом.

Алгоритмічний процес складається із самостійних етапів, кожен з яких призначений для переведення даних з одного стану в інший. З поняттям алгоритмічного процесу тісно пов'язане і поняття *обчислювального процесу* [6].

Суть алгоритмізації обчислювального процесу полягає у такому [2]:

- відокремлення автономних етапів обчислювального процесу;
- формальний запис змісту кожного з них;
- визначення порядку виконання виділених автономних етапів обчислювального процесу;
- перевірка правильності вибраного алгоритму для реалізації заданого методу обчислень.

Результати алгоритмізації обчислювального процесу систематизують у вигляді певної обчислювальної схеми, тобто деякої послідовності операцій і форм запису результатів цих операцій, яка задає алгоритм розв’язання задачі.

Базові структури алгоритмів.

Основні структури алгоритмів – це обмежений набір блоків і стандартних способів їх поєднання для виконання типових послідовностей дій. Використання кількох основних структур дає можливість будувати різноманітні алгоритми.

Всього існує чотири базових структури алгоритмів [2]:

- лінійні;
- розгалужені;
- циклічні;
- змішані.

Лінійна або послідовна без будь-яких розгалужень конфігурація алгоритму, що нагадує форму ланцюжка (рис. 1.3).

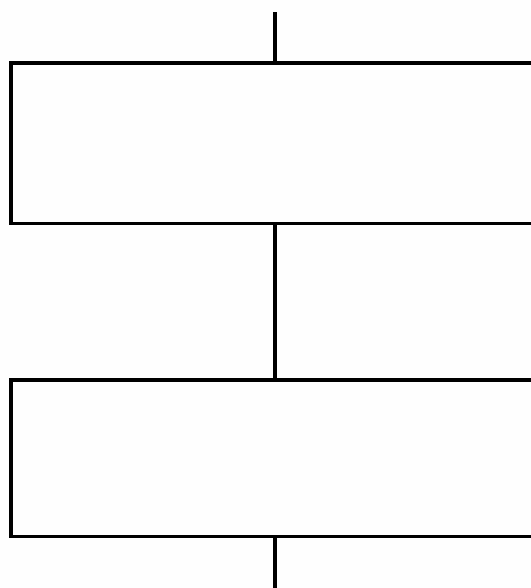


Рисунок 1.3 – Лінійна або послідовна структура алгоритму

Розгалужена конфігурація алгоритму містить у собі як звичайні послідовності, так і розпаралелені послідовності. Використовується, коли потрібно виконати ту чи іншу дію (рис. 1.4), здійснити обхід, якщо одна гілка не містить жодних дій (рис. 1.5), здійснити множинний вибір, коли умова має більше, ніж три можливі варіанти (рис. 1.6).

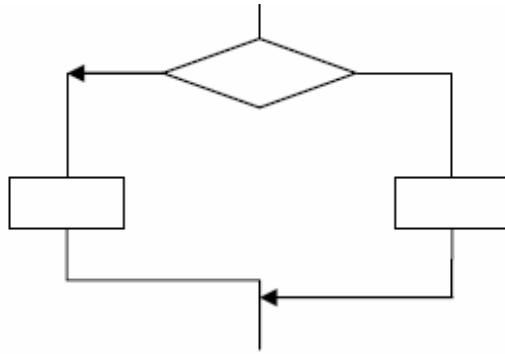


Рисунок 1.4 – Розгалужена конфігурація алгоритму, потрібно виконати ту чи іншу дію

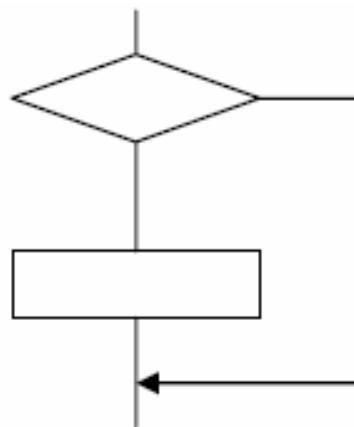


Рисунок 1.5 – Розгалужена конфігурація алгоритму, здійснено обхід, коли одна гілка не містить жодних дій

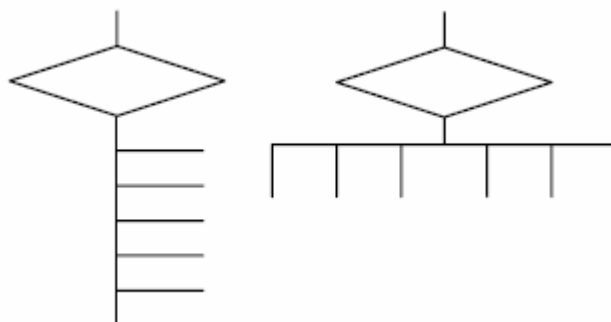


Рисунок 1.6 – Розгалужена конфігурація алгоритму, здійснено множинний вибір, коли умова має більше, ніж три можливі варіанти

Циклічна структура алгоритму використовується при необхідності виконати деякі дії кілька разів. Можливе виконання циклу До, циклу Поки, циклу за параметром (рис. 1.7).

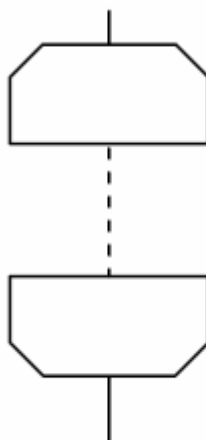


Рисунок 1.7 – Циклічна структура алгоритму

Особливістю всіх наведених структур є те, що вони мають один вхід і один вихід, тому їх можна поєднувати один з одним у будь-якій послідовності.

Досвід практичної алгоритмізації призвів до формування особливої методики структурної організації алгоритмів, використання якої зменшує ймовірність помилок у процесі розробки і запису алгоритмів, спрощує їх розуміння і модифікацію. Цю методику алгоритмізації називають *структурним підходом*. При структурному підході до конструювання алгоритмів їх ніби «збирають» із трьох основних (базових) структур.

Для подання моделей використовують такі основні форми [2]:

- інваріантна форма – це запис моделі за допомогою традиційної математичної мови;
- алгоритмічна форма – це запис моделі та обраного чисельного методу вирішення у формі алгоритму;
- аналітична форма – це запис моделі у вигляді результату аналітичного вирішення вихідних рівнянь моделі;
- схемна форма (графічна форма) – це подання моделі на деякій графічній мові. Використання таких форм можливо при наявності правил однозначного тлумачення елементів креслень та їх перекладу на мову інваріантних або алгоритмічних форм.

Моделі в алгоритмічній і аналітичній формах називають відповідно алгоритмічними й аналітичними.

Серед алгоритмічних моделей важливий клас становлять імітаційні моделі, призначені для імітації фізичних чи інформаційних процесів в об'єкті при завданні різних залежностей вхідних впливів від часу.

Вимоги до математичних моделей.

Адекватність. Модель вважається адекватною, якщо відбиває задані властивості об'єкта з прийнятною точністю. Точність визначається як ступінь збігу значень вихідних параметрів моделі й об'єкта. Очевидно, що у загальному випадку точність моделі різна в різних умовах функціонування об'єкта. Ці умови характеризуються зовнішніми параметрами. Якщо задатися граничною припустимою похибкою, то можна в просторі зовнішніх параметрів виділити область, яку називають *областю адекватності* моделі.

Універсальність. При визначенні області адекватності необхідно вибрати сукупність зовнішніх параметрів, що відбивають властивості, які враховуються у моделі. Якщо адекватність характеризується положенням і розмірами області адекватності, то універсальність моделі визначається числом і складом зовнішніх і вихідних параметрів, що враховуються у моделі.

Економічність. Характеризується витратами обчислювальних ресурсів для реалізації моделі, що включає витрати машинного часу і пам'яті.

Властивості складних систем.

1 властивість: цілісність.

Складна система розглядається як цілісна сукупність елементів, що характеризується наявністю великої кількості взаємопов'язаних і взаємодіючих між собою елементів. У дослідника існує суб'єктивна можливість розбиття системи на підсистеми, цілі функціонування яких підпорядковані спільній меті функціонування всієї системи (цілеспрямованість систем).

Цілеспрямованість інтерпретується, як здатність системи здійснювати в умовах невизначеності та впливу випадкових факторів певну поведінку (вибір поведінки) для досягнення певної мети.

2 властивість: зв'язки.

Наявність істотних стійких зв'язків (відношень) між елементами або їх властивостями. Під «зв'язками» розуміється деякий віртуальний канал, по якому здійснюється обмін між елементами та зовнішнім середовищем, речовиною, енергією, інформацією.

3 властивість: організація.

При формуванні зв'язків складається певна структура системи, а властивості елементів трансформуються у функції (дії, поведінку).

При дослідженні складних систем, зазвичай, відзначають:

- складність функції, що виконується системою і спрямованої на досягнення заданої мети функціонування;
- наявність управління, розгалуженої інформаційної мережі та інтенсивних потоків інформації;
- наявність взаємодії із зовнішнім середовищем, функціонування в умовах невизначеності та впливу випадкових факторів різної природи.

4 властивість: інтегративні якості.

Існування інтегративних якостей (властивостей), тобто таких якостей, які притаманні системі в цілому, але не властиві жодному з її елементів окремо. Наявність інтегративних якостей показує, що властивості системи хоча і залежать від властивостей елементів, але не визначаються ними повністю.

Складна система, як об'єкт моделювання, має такі особливості:

- складні системи, як правило, унікальні;
- слабка структурованість теоретичних і фактичних знань про систему.

Так як досліджувані системи унікальні, то процес накопичення та систематизації знань про них ускладнений. Під час ідентифікації складних систем присутня велика частка суб'єктивних експертних знань про систему.

Характерні особливості складних систем.

Складна система зводиться до простої сукупності елементів, роз'єднуючи складну систему на окремі частини, вивчаючи кожен з них окремо, не можна пізнати властивості системи в цілому. Опис окремих підсистем необхідно виконувати з урахуванням їх місця у всій системі в цілому, і навпаки, система в цілому досліджується виходячи з властивостей окремих підсистем.

Однією з основних рис складних систем є взаємодія виділених підсистем, необхідно враховувати результат впливу однієї підсистеми на іншу і їх взаємодію із зовнішнім середовищем.

Різноманітність підсистем і елементів системи, визначається і різноманітністю природи, і різноманітністю математичних схем, що описують функціонування різних елементів, а також одних і тих же елементів на різних рівнях вивчення. Існує необхідність дослідити систему в динаміці, з урахуванням поведінкових аспектів.

Випадковість і невизначеність факторів, що діють у досліджуваній системі. Облік цих чинників призводить до різкого ускладнення завдань і збільшує трудомісткість досліджень. Існує необхідність врахування великої кількості діючих в системі факторів.

Багатокритеріальність оцінок процесів, що протікають у системі. Неможливість однозначної оцінки (вибору єдиного узагальненого критерію) диктується такими обставинами:

- наявністю підсистем, кожна з яких має свої цілі, оцінюється за своїми локальними критеріям;
- множинністю показників (при системному підході вибирається компромісний варіант), що характеризують роботу всієї системи;
- наявністю формалізації критеріїв, що використовуються під час прийняття рішень, заснованих на практичному досвіді осіб, які приймають рішення.

При системному підході процес дослідження складної системи носить ітераційний характер. Вихідна модель ускладнюється шляхом деталізації. Однак, створення повної моделі складної системи буде настільки ж складним у вивченні, як і система. Наслідком цього є необхідність використання комплексу моделей під час аналізу системи.

Найважливіший інструмент прикладного системного аналізу – комп'ютерне моделювання. Імітаційне моделювання є найбільш ефективним і універсальним варіантом комп'ютерного моделювання в галузі дослідження і управління складними системами.

Модель – абстрактний опис системи (об'єкта, процесу, проблеми, поняття) у деякій формі, відмінній від форми їх реального існування [1].

Моделювання – один із основних методів пізнання, що є формою відображення дійсності і полягає у з'ясуванні або відтворенні тих чи інших властивостей реальних об'єктів, предметів і явищ за допомогою інших об'єктів, процесів, явищ, або за допомогою абстрактного опису у вигляді зображення, плану, карти, сукупності рівнянь, алгоритмів і програм [1].

У процесі моделювання завжди існує оригінал (об'єкт) і модель, яка відтворює (моделює, описує, імітує) деякі риси об'єкта.

Моделювання засноване на наявності різноманіття природних і штучних систем, що відрізняються як цільовим призначенням, так і фізичним втіленням схожості або подібності деяких властивостей: геометричних, структурних, функціональних, поведінкових. Ця схожість може бути повною (ізоморфізм) і частковою (гомоморфізм). Гомоморфізм – теоретична основа моделювання.

Комп'ютерне моделювання – метод вирішення завдань аналізу або синтезу складної системи на основі використання її комп'ютерної моделі [6].

До комп'ютерного моделювання відносять:

- структурно-функціональне;
- імітаційне.

Під терміном «комп'ютерна модель» розуміють умовний образ об'єкта або системи об'єктів (або процесів), описаний за допомогою взаємопов'язаних комп'ютерних таблиць, блок-схем, діаграм, графіків, малюнків, що відображають структуру та взаємозв'язки між елементами об'єкта. Комп'ютерні моделі такого виду називають *структурно-функціональними*.

Окрему програму (сукупність програм, програмний комплекс), що дозволяє за допомогою послідовності обчислень і графічного відображення їх результатів відтворювати (імітувати) процеси функціонування об'єкта, системи об'єктів за умови впливу на об'єкт різних, як правило, випадкових факторів. Такі моделі прийнято називати *імітаційними*.

Процедурно-технологічна схема побудови та дослідження моделей складних систем включає такі етапи визначення, характерні для будь-якого методу моделювання [2]:

- системи (предметна, проблемна область);
- об'єкта моделювання;
- цільового призначення моделей;
- вимог до моделей;
- форми подання;
- виду опису моделі;
- характеру реалізації моделі;
- методу дослідження моделі.

Предметна (проблемна) область визначається фізичними, хімічними, технічними, технологічними, інформаційними, біологічними, екологічними, економічними, соціальними та іншими можливими класами систем [1].

Сфера застосування імітаційних моделей обширна, імітаційне моделювання використовується для дослідження найрізноманітніших систем: економічних, виробничих, соціальних, транспортних, систем масового обслуговування, обчислювальних, інформаційних, включаючи міжнародну діяльність, проблеми розвитку міст, глобальні (світові) проблеми.

Як *об'єкт моделювання* у процесі дослідження виступає не вся система, а її «зріз» – елемент, структура, організація, функція, окремі процеси, поведінка.

Кожна модель повинна будуватися цілеспрямовано.

Цілеспрямована модель – модель повинна відображати ті суттєві властивості і сторони модельованого об'єкта, які визначені практичним завданням.

Дуже важливо правильно позначити і сформулювати проблему, чітко задати мету дослідження, проведеного за допомогою моделювання.

Для того, щоб розв'язувати задачі за допомогою електронно-обчислювальних машин, складати програми і користуватись ними, необхідна алгоритмічна культура.

Алгоритмічна культура – це частина загальної математичної культури і загальної культури мислення, яка зумовлює формування вмінь, пов'язаних з розумінням суті поняття алгоритму і його властивостей, тобто це сукупність знань, умінь і навичок, які дозволяють успішно розв'язувати задачі.

Під час розв'язання прикладних задач вибір алгоритму має певні труднощі. Одна з основних проблем, що виникає під час переходу від алгоритмів безкомп'ютерних до алгоритмів, поданих у вигляді комп'ютерної програми, пов'язана із формою їх запису, яка повинна бути зрозумілою комп'ютеру.

Алгоритм повинен відповідати таким вимогам:

- бути простим для розуміння, переводу в програмний код та налагодження;
- ефективно використовувати комп'ютерні ресурси і працювати швидко.

Вивчення алгоритмічної культури включає необхідність не тільки інтуїтивного розуміння суті поняття алгоритму та його властивостей, уявлення про можливість автоматизації тієї сфери діяльності, до якої цей алгоритм застосовується, а й уміння описувати алгоритм за допомогою певних засобів і методів, знання основних типів алгоритмів і способів їх використання.

Побудова алгоритмів і реалізація їх у вигляді програм сформували такі висновки та рекомендації [7]:

- планування етапів розробки програми (спочатку чорновий варіант алгоритму у неформальному стилі, потім псевдопрограма, далі – послідовна формалізація псевдопрограми, тобто перехід до рівня програмного коду). Така стратегія організовує і дисциплінує процес створення кінцевої програми, яка буде простою для налагодження, подальшої підтримки та супроводу;

- використання інкапсуляції. Усі процедури, що реалізують абстрактні типи даних, подаються в одному місці програмного тексту. Якщо виникне необхідність змінити реалізацію абстрактних типів даних, можна буде коректно і без особливих затрат внести будь-які зміни, оскільки всі необхідні процедури локалізовані в одному місці програми;

- використання та модифікація існуючих програм. Один із неефективних підходів до процесу програмування полягає в тому, що кожний новий проект розглядається з «нуля», без урахування вже існуючих програм. Як правило, серед програм, реалізованих на момент початку проекту, можна знайти такі, які якщо вирішують не всю поставлену задачу, то хоча б її частину. Після створення нової програми слід передбачити сфери, де її ще можна використати.

2 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

2.1 Моделювання паралельних процесів

План

1. Основні поняття імітаційного моделювання.
2. Структура імітаційного моделювання.
3. Етапи імітаційного моделювання, їх взаємозв'язок.
4. Особливості імітаційного моделювання.
5. Склад імітаційної моделі.
6. Загальна структура імітаційної моделі.
7. Класифікація імітаційних моделей за різними критеріями.
8. Переваги і недоліки імітаційних моделей.
9. Моделювання паралельних процесів.
10. Види паралельних процесів.
11. Методи опису паралельних процесів.

Імітаційне моделювання застосовується у всіх сферах діяльності людини, починаючи від моделей технічних, технологічних та організаційних систем і закінчуючи проблемами розвитку людства і всесвіту [7].

Основна цінність імітаційного моделювання полягає в тому, що в його основі лежить методологія системного аналізу. Дана методологія дозволяє здійснити дослідження проектованої або аналізованої системи за схемою операційного аналізу, що включає взаємопов'язані етапи: змістовна постановка задачі, розробка концептуальної моделі, розробка і програмна реалізація імітаційної моделі, перевірка адекватності моделі та оцінка точності результатів моделювання, планування і проведення експериментів, прийняття рішень. Такий підхід дозволяє використовувати імітаційне моделювання як універсальний метод для прийняття рішень в умовах невизначеності та для врахування в моделях важко формалізованих факторів, а також застосовувати основні принципи системного підходу для вирішення практичних задач.

Модель є абстрактним описом системи, рівень деталізації якої залежить від мети моделювання і можливості отримання вихідних даних з необхідною точністю. Включати або не включати даний елемент у модель визначає дослідник. Модель є суб'єктивною за своєю природою і відображає загальні властивості і закономірності у світі об'єктів.

Об'єкти можуть бути визначені через їх атрибути (властивості) для даного об'єкта. Успіх моделювання багато у чому залежить від того, наскільки добре дослідник може виділяти важливі елементи системи та описати взаємозв'язки між ними.

На відміну від інших видів моделювання імітаційне моделювання враховує зміну властивостей об'єктів у часі, тобто імітаційні моделі – це динамічні моделі. Вони можуть бути детермінованими або стохастичними. Облік у моделі випадкових чинників призводить до необхідності статистичної оцінки результатів моделювання.

Однією із перших мов моделювання, що полегшила процес написання імітаційних програм, була мова GPSS, створена фірмою IBM.

GPSS (General Purpose Simulating System – загально цільова система моделювання) є мовою моделювання, що призначена для побудови дискретних моделей та проведення моделювання на електронно-обчислювальних машинах.

Моделі систем на GPSS описуються у вигляді блок-схем або представлені у вигляді послідовності рядків програми, що еквівалентні блок-схемі.

Блок-схема є набором елементів з характерними блоками мови GPSS, з'єднаних між собою лініями. У систему моделювання GPSS входять спеціальні засоби для опису динамічної поведінки систем шляхом зміни станів у дискретні моменти часу. GPSS є мовою та транслятором, що містить словник і граматику, за допомогою яких можуть бути розроблені моделі систем певного типу.

У широкому розумінні імітаційне моделювання – це процес конструювання моделі реальної системи та проведення експерименту на цій моделі з метою визначення поведінки системи або оцінити (у рамках обмежень, зумовлених деяким критерієм чи сукупністю критеріїв) різні стратегії, що забезпечують функціонування цієї системи.

У вузькому розумінні імітаційне моделювання – це відтворення на електронно-обчислювальній машині реальної виробничої чи організаційної системи. У такому поданні «імітаційне моделювання» має той самий сенс, що й «машинна імітація» або «машинне моделювання».

В англomовній літературі здебільшого використовуються такі терміни:

- computer simulation (комп'ютерне моделювання);
- systems simulation (системне моделювання);
- digital simulation (цифрове моделювання).

У вітчизняній літературі розповсюджені терміни «машинна імітація», «машинне моделювання», «імітаційне моделювання».

Щоб застосувати метод імітаційного моделювання для досліджень, створюють імітаційну систему, яка містить у собі імітаційну модель, а також внутрішнє і зовнішнє математичне забезпечення.

До електронно-обчислювальної машини вводять потрібні вхідні дані і спостерігають зміни показників, які у процесі моделювання можуть аналізуватися та піддаватися статистичній обробці.

Машинну імітацію як числовий машинний метод розв'язання складних задач доцільно застосовувати за таких умов:

- непридатність або відсутність аналітичних методів розв'язання задач;
- цілковита впевненість в успішному створенні імітаційної моделі, яка адекватно описує досліджувану систему (процес), зокрема в тому, що вдасться зібрати всю необхідну інформацію про модельовану систему (процес), забезпечивши вірогідну імітацію на електронно-обчислювальній машині реальних ситуацій;
- можливість використати сам процес побудови імітаційної моделі для попереднього дослідження системи, що моделюється, з метою напрацювання рекомендацій щодо поліпшення умов її функціонування.

Відомі два способи побудови динамічних імітаційних моделей на електронно-обчислювальних машинах:

- однорідне градуювання системного (модельного) часу;
- неоднорідне градуювання системного часу.

Програму для електронно-обчислювальної машини можна розробити двома способами:

- звичайними засобами програмування із застосуванням сучасних проблемно-орієнтованих або машинно-орієнтованих мов;
- за допомогою спеціалізованих мов моделювання.

Переваги застосування імітаційного моделювання найбільш помітно виявляються у разі моделювання виробничих і технологічних процесів, а також процесів матеріально-технічного забезпечення виробництва, у логістиці, а також під час проведення бізнес-планування, екологічних і соціологічних досліджень. Важливо, що імітаційне моделювання використовується, скоріше, як спосіб для осмислення проблеми і допомагає в цьому більше, ніж простий текстовий або математичний опис проблеми. Воно дає змогу глянути на складний процес прийняття рішення більш масштабно, з погляду процесів, які відбуваються всередині системи, що моделюється.

Часто моделювання припиняють ще до того, як будуть отримані конкретні результати.

Проаналізувавши літературні джерела [2, 6, 7], зроблено такі висновки:

- імітаційне моделювання дає змогу досліджувати внутрішні взаємодії у складних системах або підсистемах у межах складної системи, а також експериментувати з ними;
- моделюючи інформаційні, організаційні впливи та впливи зовнішнього середовища, можна оцінити особливості цих впливів на поведінку системи;
- на основі знань, отриманих під час проектування імітаційної моделі, можна визначити способи вдосконалення системи, яка моделюється;
- змінюючи вхідні дані під час моделювання та спостерігаючи за вихідними даними, можна виявити, які змінні найбільш важливі та як вони взаємодіють;
- імітаційне моделювання можна використовувати як метод для покращення рішень, отриманих під час аналітичного аналізу, а також для перевірки аналітичних рішень;
- імітаційне моделювання можна використовувати для проведення експериментів з новими проектами або стратегіями їх упровадження, щоб заздалегідь спрогнозувати результати;
- імітаційне моделювання можна застосовувати для визначення вимог, яким має відповідати пристрій або система;
- імітаційні моделі можна використовувати для навчання операторів складних технологічних процесів без зайвих затрат на придбання обладнання, яке може пошкодитись, і запобігаючи нещасним випадкам;
- для імітаційного моделювання можна використовувати засоби анімації, які дають змогу спостерігати за операціями, що моделюються;
- сучасне виробництво настільки складне, що взаємозв'язки в ньому можна інтерпретувати тільки шляхом проведення імітаційного моделювання.

Розглянемо детальний аналіз дій, що виконуються на етапі побудови імітаційної моделі.

Послідовність складання імітаційної моделі передбачає такі кроки [1]:

- визначення задачі та її аналіз;
- визначення вимог до інформації;
- збирання інформації;
- висування гіпотез та прийняття припущень;
- встановлення основного змісту моделі;
- визначення параметрів, змінних і критеріїв ефективності;
- опис концептуальної моделі й перевірка її вірогідності;
- побудова логічної структурної схеми (блок-схеми).

На етапі постановки задачі імітаційного моделювання необхідно [7]:

- перевірити існування задачі та необхідність машинного моделювання;
- дослідити задачу за матеріалами літературних джерел;
- дати чітке формулювання задачі;
- вибрати методику розв’язання;
- перевірити ресурси, необхідні для моделювання задачі на комп’ютері;
- визначити масштабність задачі та можливість її поділу на підзадачі;
- визначити послідовності розв’язання підзадач.

На етапі роботи, пов’язаної з аналізом задачі моделювання виконуються такі функції [8]:

- обираються критерії оцінки процесу функціонування системи, що досліджується;
- виділяються системи ендогенних та екзогенних змінних моделі;
- обираються можливі методи ідентифікації;
- виконується попередній аналіз наступних двох етапів моделювання.

На цьому етапі виконуються роботи, пов’язані з підготовкою та реалізацією імітаційної моделі на комп’ютері. Розробляється логічна схема моделі, яка потім перетворюється у програму. Подальша формалізація концептуальної моделі відбувається на етапі розробки структури імітаційної моделі, але слід мати на увазі, що відображення структури моделі залежить від обраних засобів програмування.

Програмна реалізація імітаційної моделі створюється за допомогою [5]:

- алгоритмічних мов загального призначення;
- спеціалізованих мов моделювання;
- пакетів прикладних програм для моделювання;
- засобів автоматизації програмування імітаційних моделей;
- діалогових і візуальних систем моделювання;
- інтелектуальних систем моделювання.

Програмну реалізацію імітаційної моделі рекомендується будувати за модульним принципом. Це дає змогу удосконалювати модель за допомогою ітераційного методу, додаючи до неї модуль за модулем. Під час процесів налагоджування та експериментування окремі модулі може бути замінено або змінено, що не призведе до істотних змін в усій моделі.

Структура програми моделі має відповідати структурі імітаційної моделі. Така побудова програми робить її наочною та полегшує її налагодження. Кожний модуль програми має супроводжуватись коментарем. Програмування та налагодження моделі доцільно проводити поетапно.

Для оцінювання правильності функціонування програмної реалізації імітаційної моделі проводяться пробні експерименти (тестування моделі), в яких широко використовуються налагоджувальні засоби вибраної системи моделювання. Більшість мов моделювання має засоби, які дають змогу слідкувати за рухом процесів у моделі, завдяки чому за різних початкових умов можна переконатися у тому, що модель працює так, як було задумано.

Типова помилка під час налагодження моделі пов'язана з неузгодженістю пропускну здатності окремих елементів системи, тобто процеси надходять у деякі елементи моделі швидше, ніж вони встигають обслуговуватись. Через це доцільно на деяких ділянках моделі, в яких можуть нагромаджуватись процеси, задавати обмежувальні умови на довжину черги.

Після закінчення налагодження функціонування програмної реалізації імітаційної моделі необхідно перевірити її працездатність в усьому діапазоні змін вхідних змінних. Усі значення змінних у моделі має бути зведено до вибраної одиниці модельного часу. Для остаточного тестування моделі на контрольних прикладах необхідно залучати тих людей, які не брали участі в програмуванні моделі або майбутніх її користувачів.

Більшість комп'ютерних архітектур і мов програмування орієнтовані на послідовне виконання операторів програми.

Опис динаміки системи, її поведінки, складає основу будь-якої імітаційної моделі. Як початкові дані для вирішення цієї задачі використовуються результати, отримані на етапі розробки концептуальної моделі системи, до них відносяться:

- визначення належності системи, що моделюється, відомому класу;
- опис робочого навантаження системи;
- вибір рівня деталізації подання системи в моделі і її декомпозиції.

Усі наступні дії дослідника по створенню моделі можуть бути віднесені до етапу її формалізації, що у загальному випадку припускає:

- вибір методу відображення динаміки системи;
- формальний (математичний) опис випадкових факторів, що підлягають обліку в моделі;
- вибір механізму зміни і масштабу модельного часу.

Визначимо основні поняття: «робота», «процес», «подія», «транзакт».

Робота (активність) – це одинична дія системи по обробці (перетворенню) вхідних даних. У залежності від природи системи, що моделюється, під вхідними даними можуть розумітися інформаційні дані чи які-небудь матеріальні ресурси [1].

Робота характеризується часом виконання та споживаними ресурсами.

Під *процесом* розуміють логічно зв'язаний набір робіт [1].

Будь-який процес характеризується сукупністю статичних і динамічних характеристик.

До статичних характеристик процесу відносяться:

- тривалість;
- результат;
- споживані ресурси;
- умови запуску (активізації);
- умови зупинки (переривання).

У загальному випадку статичні характеристики процесу не змінюються в ході його реалізації, однак, при необхідності кожна з них може бути подана у моделі як випадкова величина, розподілена за заданим законом.

Динамічною характеристикою процесу є його стан (активний чи знаходиться у стані очікування).

Моделювання у термінах процесів виконується у тих випадках, коли система оцінюється за будь-якими тимчасовими показниками або з погляду споживаних ресурсів. У випадку, коли модель будується з метою вивчення причинно-наслідкових зв'язків, властивих системі, динаміку системи доцільно описувати у термінах подій.

Подія – це миттєва зміна елемента системи чи стану системи в цілому [1].

Подія характеризується:

- умовами (чи законом) виникнення;
- типом, що визначає порядок обробки даної події;
- нульовою тривалістю.

Події підрозділяють на дві категорії:

- події, що керують ініціалізацією процесів (чи окремих робіт у процесі);
- події зміни станів (елементів системи або системи в цілому).

Механізм подій використовується як основа побудови моделей, призначених для дослідження причинно-наслідкових зв'язків у системах за відсутності тимчасових обмежень.

Транзакт – це подія, що надходить із зовні на вхід системи і підлягає обробці [7].

У даний час існують програмно-апаратні комплекси, що дозволяють організувати паралельне виконання різних частин одного обчислювального процесу. Для програмування таких систем необхідна спеціальна підтримка з боку засобів програмування, зокрема, мов програмування.

Деякі мови загального призначення містять у собі елементи підтримки паралелізму. Слід зауважити, що програмування паралельних систем потребує спеціальних прийомів.

У системах з паралелізмом на перший план висуваються питання якісного характеру:

- як взаємодіють окремі компоненти системи?
- чи існують у системі потенційно «вузькі» місця?
- чи можуть у системі виникнути порушення функціонування, збої та як можна їх локалізувати?
- чи можна спростити систему, замінивши її окремі компоненти?
- чи може система виконувати ті функції, для яких призначена?

Дослідника цікавлять структурні характеристики і властивості.

У таких випадках модель системи повинна бути структурно подібною самій системі. Це означає, що модель можна будувати з окремих складових частин як і саму систему. Глобальні функції і поняття можуть визначатися на основі локальних, відповідних компонентів, підсистем і підпроцесів, а зв'язки і відношення між фрагментами моделі повинні бути подібні зв'язкам і відношенням між фрагментами системи.

Для кращого розуміння та подання паралельних процесів розглянемо їх класифікацію. У межах функціональної моделі класифікувати можна як окремі процеси, так і взаємозв'язки між ними. У першому випадку класифікація полягає у визначенні категорії, до якої належить процес або підпроцес у межах конкретної моделі, у другому випадку класифікація полягає у визначенні категорії об'єктів, які передаються між процесами під час їх взаємодії.

Асинхронний паралельний процес – такий процес, стан якого не залежить від стану іншого паралельного процесу. Приклад асинхронних паралельних процесів, що протікають у рамках однієї системи: підготовка і проведення кампанії з продажу квартир та робота на будівельному майданчику.

Синхронний паралельний процес – такий процес, стан якого залежить від стану взаємодіючих з ним паралельних процесів. Приклад синхронного паралельного процесу: робота на будівельному майданчику і доставка матеріалів.

Процес може бути синхронним відносно певного активного паралельного процесу і асинхронним відносно іншого. При роботі обчислювальної мережі за технологією «клієнт-сервер» кожен із вузлів мережі синхронізує свою роботу з роботою сервера, але не залежить від роботи інших вузлів.

Підлеглий паралельний процес – створюється і управляється іншим процесом більш високого рівня.

Незалежний паралельний процес – не є підлеглим ні для одного з інших процесів.

Спосіб організації паралельних процесів у системі залежить від фізичної сутності цієї системи.

Розробка і використання кожної імітаційної моделі припускає її програмну реалізацію і дослідження із застосуванням обчислювальних систем. Для реалізації моделей, що імітують паралельні процеси, у деяких випадках застосовують механізми, які використовуються під час виконання паралельних обчислень.

Реалізація паралельних процесів обчислювальних систем має такі особливості:

- обчислювальні процеси можуть бути паралельними тільки в багатопроцесорних обчислювальних системах чи обчислювальних мережах;
- часто паралельні процеси використовують загальні ресурси, тому навіть асинхронні паралельні процеси у межах однієї обчислювальної системи змушені погоджувати свої дії під час взаємодії з загальними ресурсами;
- в обчислювальних системах додатково використовується ще два види паралельних процесів:

- 1) батьківський паралельний процес;
- 2) дочірній паралельний процес.

Їх особливість полягає у тому, що батьківський процес не може бути завершений, поки не завершаться всі його дочірні процеси.

Таким чином, для організації взаємодії паралельних процесів у обчислювальних системах використовуються три основних підходи:

- на основі «взаємного виключення»;
- на основі синхронізації за допомогою сигналів;
- на основі обміну інформацією.

«Взаємне виключення» припускає заборону доступу до загальних ресурсів (загальних даних) для всіх паралельних процесів, крім одного, на час його роботи з цими ресурсами (даними).

Синхронізація передбачає обмін сигналами між двома чи більше процесами за встановленими правилами. Такий «сигнал» розглядається як деяка подія, що викликає відповідні дії у процесі, що його одержав.

Іноді виникає необхідність передавати від одного паралельного процесу іншому більш докладну інформацію, ніж просто «сигнал-подія», у цьому випадку процеси узгоджують свою роботу на основі обміну інформацією.

2.2 Основні поняття та принципи побудови моделей, заснованих на знаннях

План

1. Методи штучного інтелекту в імітаційному моделюванні.
2. Застосування мереж Петрі.
3. Модифіковані мережі Петрі та сучасні методи моделювання паралельних процесів у складних системах.
4. Елементи та властивості мережі Петрі.
5. Правила спрацювання переходів.
6. Граф досяжних маркувань.
7. Об'єктно-орієнтоване візуальне моделювання.

У багатьох роботах [1, 4, 6 – 8] відзначається, що графові моделі є найбільш зручним та ефективним засобом опису і дослідження паралельних структур та процесів. Існує декілька формалізмів, в основі яких лежать графові моделі, найвідомішими серед них є схеми паралельних програм Карпа-Мілнера, А-програми Котова-Нарін'яни, біологічні графи, оператори Хоара.

Теорія мереж Петрі є добре відомим і популярним формалізмом, призначеним для роботи з паралельними та асинхронними системами. Заснована на початку 60-х років німецьким математиком К. А. Петрі, у даний час вона містить велику кількість моделей, методів і засобів аналізу, що мають велику кількість додатків практично у всіх галузях імітаційного моделювання.

У 1962 році мережі Петрі вперше були описані Карлом Адамом Петрі в докторській дисертації «Зв'язок автоматів» (основи мереж Петрі).

У 1968 році робота Карла Адама Петрі привернула увагу Хольта А. В. і співробітників проекту Теорія інформаційних систем (розвинена початкова теорія мереж Петрі, запропоновані позначення та подання мереж Петрі).

У 1970-1980 роках відбулося розширення вивчення мереж Петрі.

У 1981 році Джеймс Пітерсон написав книгу «Теорія мереж Петрі і моделювання систем».

Мережі Петрі – інструмент дослідження систем, що застосовується виключно у моделюванні. Аналіз мереж Петрі допоможе отримати важливу інформацію про структуру та динамічну поведінку системи, що моделюється. Мережі Петрі моделюють системи, які мають взаємодіючі компоненти.

Мережі Петрі – математичний апарат для моделювання динамічних дискретних систем [1].

Розвиток теорії мереж Петрі здійснювався у двох напрямках.

Формальна теорія мереж Петрі займається розробкою основних засобів, методів і понять, необхідних для застосування мереж.

Прикладна теорія мереж Петрі пов'язана із застосуванням мереж Петрі до моделювання систем, їхнього аналізу і глибокого проникнення у структуру системи як результату моделювання.

Мережа Петрі є математичною моделлю у термінах «умова – подія».

Події – це дії, що мають місце в системі. Виникненням подій управляє стан системи. Стан системи може бути описано множиною умов.

Умова – це предикат або логічний опис стану системи. Умова може приймати значення «істина» або «хибне».

Оскільки події – це дії, то вони можуть відбуватися, щоб подія відбулася, необхідно виконати відповідні умови, що називаються *передумовою події*.

Виникнення події може викликати порушення передумов і привести до виконання інших умов, постумови.

Виконання мережі Петрі розглядається як послідовність дискретних подій. Якщо в якийсь момент часу дозволено більше одного переходу, то будь-який із кількох можливих переходів може стати «наступним» запущеним.

Вибір запущеного переходу здійснюється недетермінованим чином, тобто випадково. Ця особливість мережі Петрі відображає той факт, що у реальній життєвій ситуації, де кілька подій відбувається одночасно, виникаючий порядок подій неоднозначний, може виникнути будь-яка з множини послідовностей подій. Однак, частковий порядок появи подій – єдиний.

Мережа Петрі складається із чотирьох елементів [1]:

- множини позицій P (схематично позначаються колами);
- множини переходів T (позначаються рисками);
- вхідної функції I ;
- вихідної функції O .

Мережа Петрі C є четвіркою [1]:

$$C = (P, T, I, O), \quad (2.1)$$

де $P = \{p_1, p_2, \dots, p_n\}$ – кінцева множина позицій, $n \geq 0$;

$T = \{t_1, t_2, \dots, t_m\}$ – кінцева множина переходів, $m \geq 0$;

I – це вхідна функція, що відображає перехід у комплекти позицій;

O – це вихідна функція, що відображає перехід у комплекти позицій.

Вхідна та вихідна функції пов'язані з переходами і позиціями.

Орієнтовані дуги (стрілки) з'єднують позиції і переходи, при цьому деякі дуги спрямовані від позицій до переходів, а інші – від переходів до позицій, як показано на рисунку 2.1.

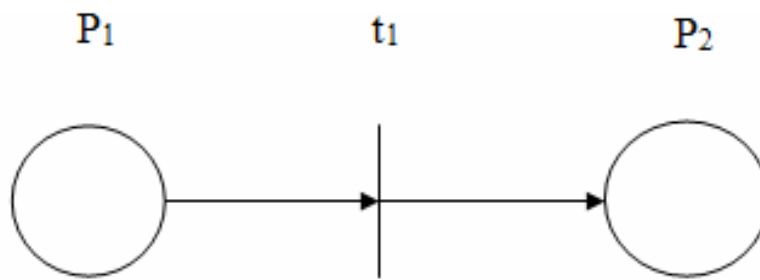


Рисунок 2.1 – Поєднання позицій та переходу

Маркірування μ – присвоєння маркерів позиціям мережі Петрі.

Маркер – примітивне поняття мережі Петрі. Маркери знаходяться у позиціях і управляють виконанням переходів мережі.

Маркірована μ мережа Петрі – сукупність структури мережі Петрі (2.1) та маркірування μ , що може бути записана у вигляді [8]:

$$M = (P, T, I, O, \mu). \quad (2.2)$$

Множина всіх маркірувань мережі, що має n позицій, є множиною всіх n векторів.

Для того, щоб подати систему засобами мереж Петрі потрібно [1]:

- виділити події, що виникають в системі, і поставити у відповідність кожній події перехід мережі Петрі;
- з'ясувати умови, при яких виникає кожна з подій, і поставити у відповідність кожній умові позицію мережі Петрі;
- визначити кількість маркерів у позиції мережі Петрі, що символізує виконання умови;
- з'єднати позиції та переходи відповідно до логіки виникнення подій у системі: якщо умова передуює виконанню події, то з'єднати в мережі Петрі відповідну позицію з відповідним переходом; якщо умова є наслідком виконання події, то з'єднати в мережі Петрі відповідний перехід з відповідною позицією;
- з'ясувати зміни, які відбуваються у системі під час здійснення кожної події, і поставити у відповідність змінам переміщення певної кількості маркерів із позицій у переходи та з переходів у позиції;
- визначити числові значення часових затримок у переходах мережі Петрі;
- визначити стан мережі Петрі на початку моделювання.

Мережа Петрі виконується за допомогою запуску переходів.

Перехід може запускатися тільки в тому випадку, коли він дозволений.

Перехід називається дозволеним, якщо кожна з його вхідних позицій має число маркерів рівне числу дуг з позиції в перехід (рис. 2.2), інакше (рис. 2.3).

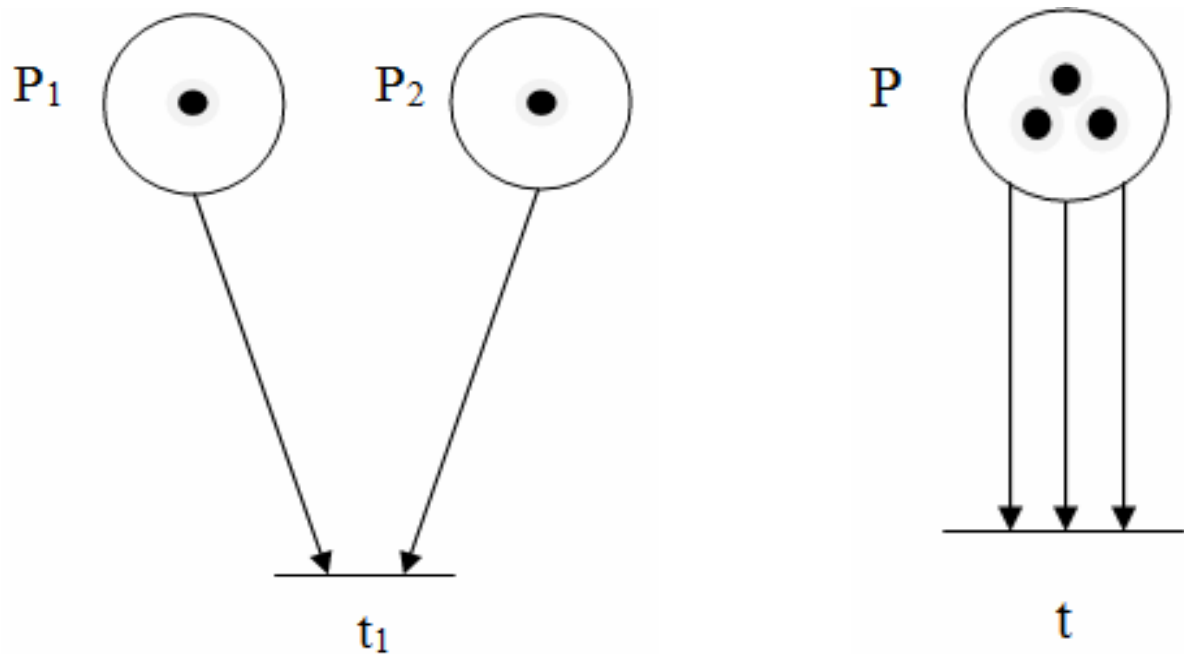


Рисунок 2.2 – Перехід дозволено

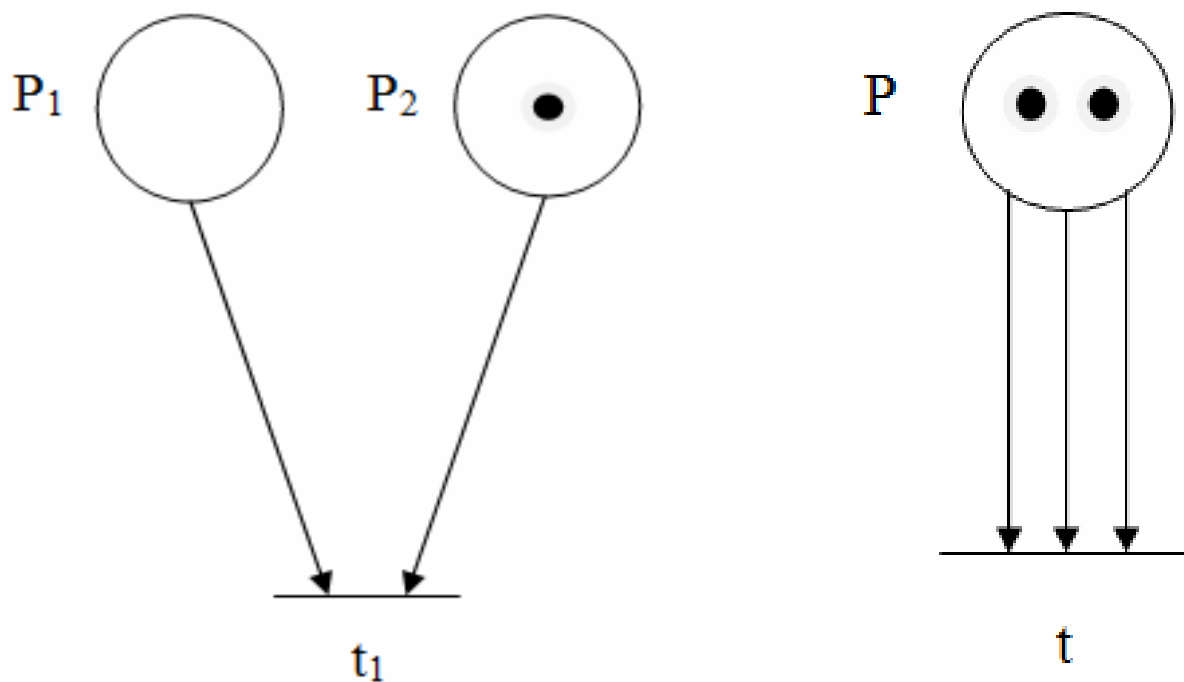


Рисунок 2.3 – Перехід не дозволено

Наприклад, перехід t_1 при $I(t_1) = \{P_1\}$ і $O(t_1) = \{P_2, P_3\}$ дозволений кожного разу, коли в P_1 буде хоча б один маркер (рис. 2.4).

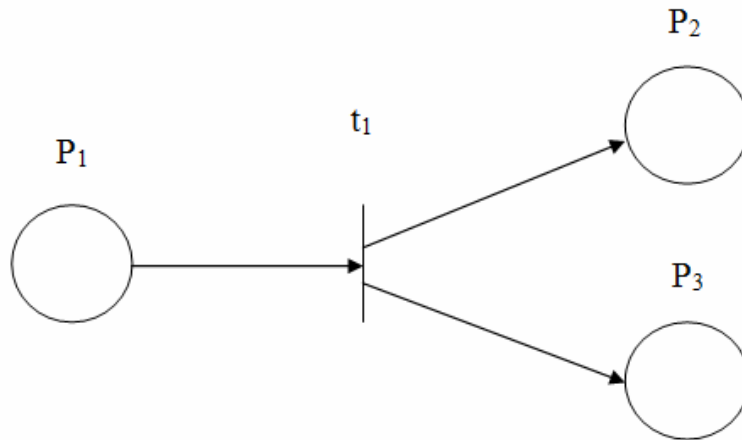


Рисунок 2.4 – Перехід t_1 дозволений, коли в P_1 буде хоча б один маркер

Перехід t_j у маркірованій мережі Петрі з маркіруванням μ може бути запущено кожного разу, коли він дозволений. У результаті запуску дозволеного переходу t_j утворюється нове маркірування μ^* .

Граф G мережі Петрі – дводольний орієнтований мультиграф [1]

$$G = (V, A), \quad (2.3)$$

де $V = \{v_1, v_2, \dots, v_s\}$ – множини вершин;

$A = \{a_1, a_2, \dots, a_r\}$ – комплект спрямованих дуг,

$$a_i = \{v_j, v_k\}, \quad (2.4)$$

де v_j, v_k належать V .

Мережа Петрі – мультиграф, він допускає існування кратних дуг від однієї вершини до іншої, так як дуги направлені, то це орієнтований мультиграф, який є дводольним.

Однією з особливостей мереж Петрі та їх моделей є паралелізм або одночасність. Мережі Петрі є ідеальними для моделювання систем із розподіленим управлінням, в яких кілька процесів виконуються одночасно.

Важливою особливістю мереж Петрі є їх асинхронна природа. У мережі Петрі відсутній вимір часу або плин часу.

Виконання мережі Петрі розглядається як послідовність дискретних подій. Не примітивними називаються події, тривалість яких відмінна від нуля.

Якщо два переходи можуть бути запущені у будь-якому порядку, то таку властивість називають недетермованість і неодночасність (рис. 2.5).

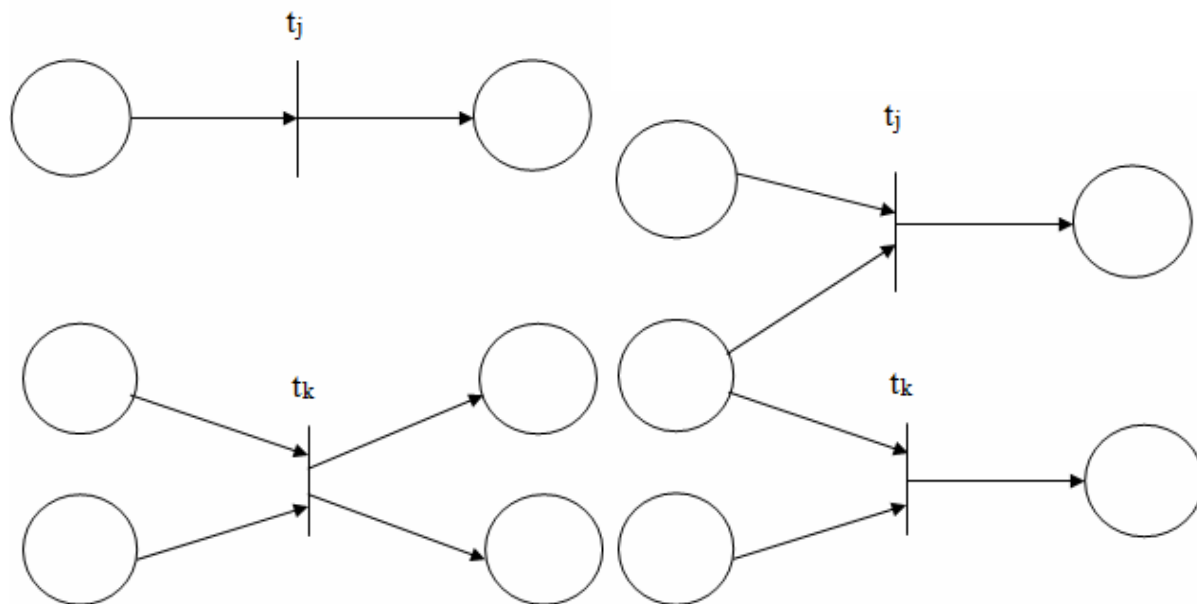


Рисунок 2.5 – Недетермованість і неодночасність

Якщо перехід t_k може бути запущений у будь-якому порядку, але обов'язково за допомогою маркерів в обох позиціях, то таку властивість називають одночасністю.

Якщо одночасне виконання переходів характеризується неможливістю одночасного запуску, то переходи t_i та t_k знаходяться в конфлікті, так як запуск одного з них видаляє маркер із позиції та завершує другий перехід, ця ситуація називається конфліктом, а в модельованих системах відображає боротьбу за спільні ресурси.

Існують певні області, в яких мережі Петрі є ідеальним інструментом для моделювання, це області, в яких події відбуваються синхронно і незалежно. Однією із таких областей є використання мереж Петрі для моделювання апаратного та програмного забезпечення електронно-обчислювальних машин та інших складних систем.

Розроблені моделі дозволяють описати і проаналізувати:

- механізми взаємодії процесів;
- тимчасові відношення між виконаними процесами;
- абсолютні години;
- управління виключними ситуаціями, яке визначається порушеннями.

Практичне застосування мереж Петрі під час проектування та аналізу систем [1]:

– мережі Петрі розглядаються як допоміжний елемент аналізу. Для побудови системи використовуються загальноприйняті методи проектування, будується мережа Петрі і аналізується модель (рис. 2.6);

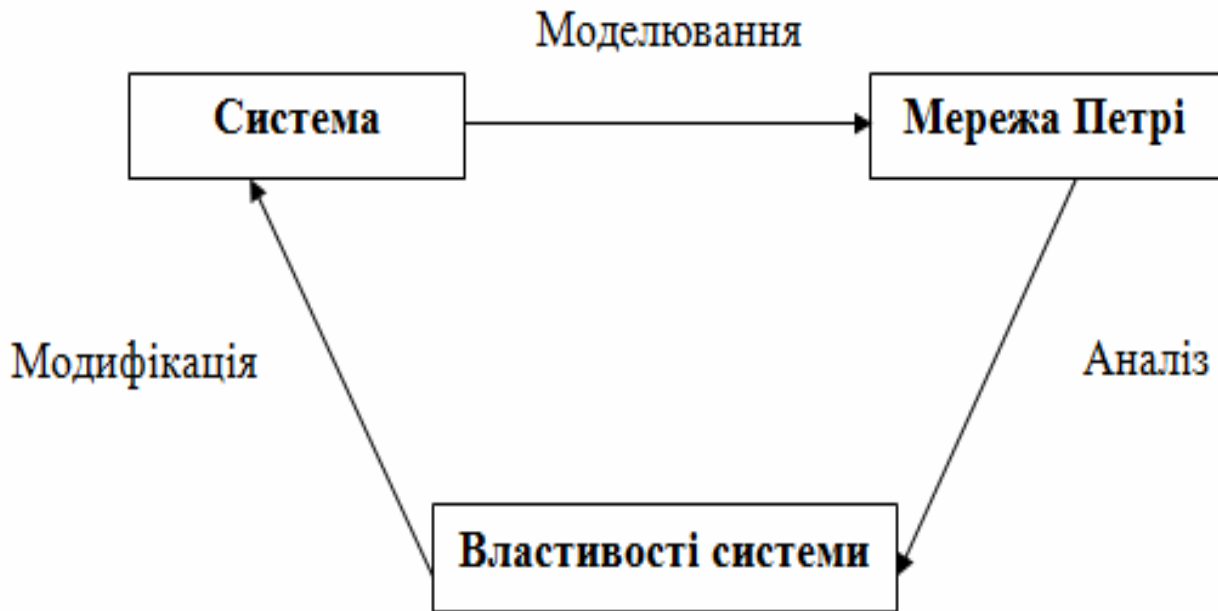


Рисунок 2.6 – Практичне застосування мереж Петрі

– процес проектування і визначення характеристик проводиться у термінах мереж Петрі. Методи аналізу застосовуються тільки для створення проекту мережі Петрі. Завдання полягає у перетворенні мережі Петрі у реальну робочу систему.

Побудовані моделі дозволяють здійснювати такі операції:

- статичний аналіз системи;
- динамічний аналіз системи для конкретного маркірування мережі;
- імітаційне моделювання системи з побудовою графіків руху маркерів щодо позиції мережі в системному часі та в реальному часі шляхом завдання затримок часу, що відображають тривалість реальних операцій.

Кількість маркерів у позиціях складає вектор, який називається *маркіруванням*. На графічному поданні мережі Петрі зображується тільки початкове маркірування. У процесі функціонування мережі Петрі маркірування змінюється.

Умовою запуску переходу є наявність маркерів у його вхідних позиціях у кількості рівній кількості зв'язків.

Якщо умова запуску переходу виконана, то здійснюється *запуск переходу*: з усіх вхідних позицій маркери видаляються у кількості рівній кількості зв'язків, а в усі вихідні позиції маркери додаються у кількості рівній кількості зв'язків.

Наприклад, на рисунку 2.7 умова запуску переходу t_1 виконана, оскільки в позиції P_1 є один маркер і в позиції P_2 є один маркер, а умова запуску переходу t_2 – не виконана, так як у позиції P_2 немає трьох маркерів. Після запуску переходу t_1 у позиції P_1 маркер залишиться, у позиції P_2 стане один маркер, а в позиції P_3 з'явиться два маркери.

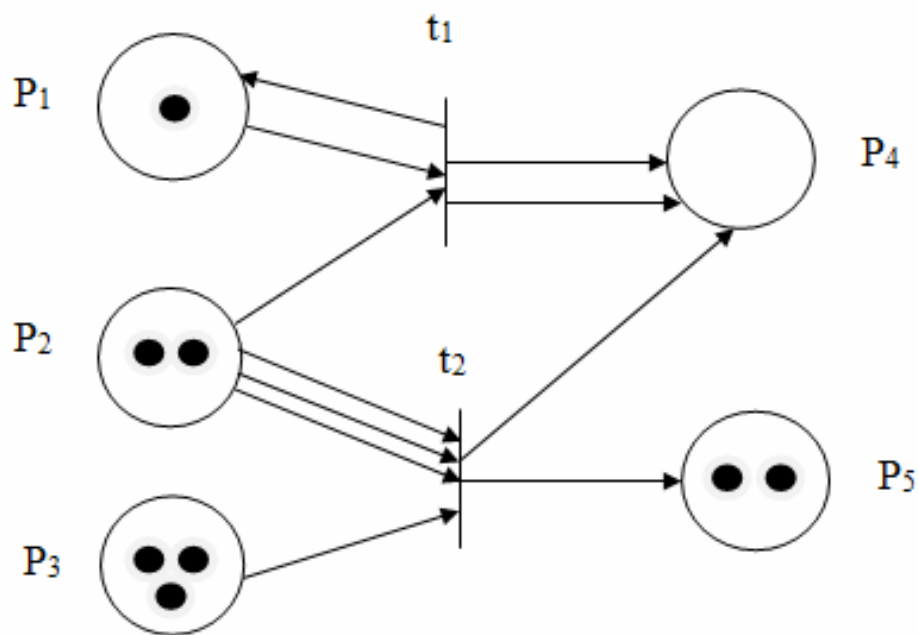


Рисунок 2.7 – Приклад застосування мереж Петрі

Взагалі, позиції та переходи можна зображати у будь-якому порядку і позначати будь-якими буквами. Головна мета побудови мереж Петрі – читабельність. Час, витрачений на подання мережі Петрі у найбільш зрозумілому вигляді, окупиться меншою кількістю помилок під час її реалізації.

Під час зображення мережі Петрі маркери у позиціях розставляються так, як вони розташовані на початку роботи моделі. Розташування маркерів змінюється, як результат функціонування моделі. Зрозуміло, що для того, щоб розпочався процес функціонування моделі, початкове розташування маркерів у позиціях повинно дозволяти запуск одного або декількох переходів.

Для переходу може бути задана часова затримка. При виконанні умови запуску переходу з усіх вхідних позицій маркери видаляються, а додаються маркери у вихідні позиції тільки після того, як сплине заданий час затримки.

Мережу Петрі, в якій задані часові затримки, називають *мережею Петрі з часовими затримками*. Імітаційні моделі складних систем складають на основі саме мереж Петрі з часовими затримками.

Процес функціонування звичайної мережі Петрі полягає у повторюваному послідовному перегляді усіх переходів мережі Петрі і виконанні таких дій:

- якщо умова запуску переходу виконана, здійснюється вхід маркерів у перехід і запам'ятовується момент виходу маркерів із переходу;
- обирається перехід, для якого момент виходу маркерів із переходу найменший, здійснюється вихід маркерів з обраного переходу;
- вихід маркерів із переходу змінює маркірування мережі Петрі, тому знову перевіряється умова запуску переходу. Процес продовжується поки запускається хоча б один перехід або поки буде вичерпано час на спостереження мережі Петрі.

Види мереж Петрі:

- тимчасова мережа Петрі;
- стохастична мережа Петрі;
- функціональна мережа Петрі;
- кольорова мережа Петрі;
- інгібіторна мережа Петрі;
- ієрархічна мережа Петрі;
- WF-мережа.

Мережа Петрі є зручним і потужним засобом моделювання асинхронних, паралельних, розподілених і недетермінованих процесів, що дозволяють наочно представити динаміку функціонування систем та їх елементів.

Властивості мереж Петрі:

- обмеженість (число маркерів у будь-якій позиції мережі не може перевищити деяке значення k . Позиція є k -обмеженою, якщо кількість маркерів у ній не перевищує ціле число k);
- безпека (позиція мережі Петрі є безпечною, якщо число маркерів у ній не перевищує 1. Мережа Петрі безпечна, якщо безпечна кожна її позиція);
- збереженість (деякі маркери є ресурсами, вони ніколи не створюються і не знищуються, тобто загальне число маркерів є постійною величиною);
- активність (можливість спрацювання будь-якого переходу під час функціонування змодельованого об'єкта);
- досяжність (можливість переходу мережі з одного заданого стану в інший);
- покриття (можливість досягнути стан, у який потрібно перейти).

Простір станів мережі Петрі.

Стан мережі визначається її маркіруванням. Простір станів мережі Петрі, що володіє n позиціями, є множиною всіх маркірувань.

Зміна у стані, викликана запуском переходу, визначається функцією δ , яка називається функцією наступного стану. Перехід може бути запущений тільки тоді, коли він дозволений. Функція не визначена, якщо перехід не дозволено у даному маркіруванні, якщо перехід дозволено, то маркірування отримують у результаті видалення маркерів з входів і додавання їх у виходи.

Дерево досяжності.

Дерево досяжності є множиною досяжності мережі Петрі.

Дерево досяжності можна використовувати для вирішення завдань безпеки, обмеженості, збереження і покриття, але у загальному випадку його не можна використовувати для вирішення завдань досяжності та активності.

Правила побудови мережевих моделей:

- напрямок стрілок на мережевій моделі слід зображати зліва направо;
- форма мережевої моделі повинна бути простою, без зайвих перетинів, більшість процесів слід зображати горизонтальними лініями;
- під час виконання паралельних процесів, коли одна подія є початком двох робіт і більше, що закінчуються іншою подією, для $n - 1$ роботи вводиться додаткова подія і логічний зв'язок;
- якщо ті чи інші процеси починаються після часткового виконання попереднього процесу, то такий процес слід розбити на частини, при цьому кожна частина процесу на графіку вважається самостійною;
- кодування подій;
- укрупнення мереж (правила):
 - а) група робіт на мережевому графіку зображується як один процес, якщо у цих робіт є одна початкова і одна кінцева подія;
 - б) укрупнювати в один процес слід такі роботи, які закріплені за одним виконавцем;
 - в) до укрупненої мережі не можна вводити нові події, яких не було на більш детальному графіку до укрупнення;
 - г) найменування процесів в укрупненому графіку повинно бути пов'язане з найменуванням укрупнених робіт;
 - д) номери подій, які зберігаються в укрупненій події, повинні бути такими як у детальному графіку;
- на мережевій моделі (графіку) не повинно бути замкнутих контурів, «хвостів» та глухих кутів.

Методологія об'єктно-орієнтованого моделювання.

Стандартною нотацією для моделювання великих інформаційних систем на базі об'єктно-орієнтованої методології є уніфікована мова моделювання UML (Unified Modeling Language), що підтримується багатьма CASE-засобами. Найчастіше використовують уніфіковану мову моделювання – Rational Rose.

Авторами мови UML є Грейді Буч (Grady Booch), Джеймс Рамбо (James Rumbaugh) та Айвар Якобсон (Ivar Jacobson).

Уніфікована мова моделювання UML – це графічна мова для специфікації, візуалізації, конструювання і документування програмних систем.

Специфікація – це декларативний опис того, як усе побудовано або працює. UML надає достатньо формальні та універсальні засоби для специфікації усіх можливих артефактів, що дає змогу знизити ризик неоднозначного сприйняття специфікації.

Візуалізація – представлення інформації у графічній формі, придатній для сприйняття людиною. Часто моделювання є єдиним засобом, який дає змогу уявити систему загалом як одне ціле. Проблема полягає в обмеженому сприйнятті людиною складних сутностей. Моделювання передбачає розбиття складної системи на прості складові та окремо розглядає кожну з цих складових, а також дає змогу створювати високорівневі моделі всієї системи, відкидаючи деталі, несуттєві для цього рівня абстракції.

Конструювання – отримання набору програмних модулів. Розроблені моделі системи утворюють деякий базовий каркас, на основі якого можна будувати систему. Сучасні CASE-засоби дають змогу автоматизувати конструювання програмного коду на підставі розроблених моделей.

Документування проектних рішень. Для підтримки та розвитку програмних продуктів потрібна вичерпна та якісна документація. Моделювання дає змогу одержати документи, які визначають високорівневу організацію системи. Такі документи необхідні для початкового ознайомлення із системою.

За допомогою UML можна розробити детальний план такої системи, який відображатиме концептуальні елементи системи (системні функції та бізнес-процеси), особливості її реалізації (класи, схеми баз даних, програмні компоненти багаторазового використання).

Сьогодні UML є загальновизнаним стандартом, який використовує більшість розробників системного та прикладного програмного забезпечення. Знань UML вимагають не лише від системних аналітиків та проектувальників, але й від звичайних програмістів і тестувальників програмного забезпечення.

Постійно збільшується ринок UML-орієнтованих інструментальних засобів, призначених для автоматизації процесу розробки програм.

Серед головних властивостей UML можна виділити такі:

- UML – це мова моделювання, яку використовують у контексті деякого процесу розробки програмних засобів;
- UML – це функціонально завершена мова, яка забезпечує повний цикл моделювання програмного забезпечення, починаючи від формування концепції майбутньої системи і завершуючи питаннями програмної реалізації системи;
- UML – це об'єктно-орієнтована мова, яку ефективно можна застосовувати у контексті об'єктно-орієнтованих методів аналізу та проектування;
- UML – це формальна мова, яка дає змогу будувати завершені моделі, яким властива однозначна інтерпретація. Наслідком формалізації мови є можливість її інтерпретації не лише людьми, але і машинами;
- UML – це мова візуалізації, орієнтована на подання моделей програмних систем переважно в графічній формі. UML припускає додавання до моделі текстової інформації з метою додаткової конкретизації деталей;
- UML – це стандартна мова, яка гарантує вільне розуміння та поширення UML-моделей між різними розробниками;
- UML – це універсальна мова, яка з однаковим успіхом придатна для моделювання як великих, так і малих програмних систем;
- UML – це незалежна мова, яка не накладає обмежень на мови програмування для реалізації моделей та може бути сумісною з будь-якою об'єктно-орієнтованою мовою.

Використання UML найефективніше в інформаційних системах масштабу міста, підприємства, банківських і фінансових установах, на транспорті, у торгівлі, науці.

Концептуальна модель (або метамодель) UML складається з таких частин:

- архітектурного базису;
- правил мови;
- загальних механізмів мови.

Важливим аспектом успішного створення складної інформаційної системи є використання методології розробки проекту, у рамках якої вводяться етапи роботи, ставляться задачі аналітикам, проектувальникам, програмістам, тестувальникам, системним інтеграторам у рамках методології Rational Unified Process (RUP). Методологія RUP базується на побудові системи канонічних діаграм – одиничних описів фрагментів системи. Діаграми візуалізують різні аспекти системи, у кожній діаграмі є своя мета і своя категорія користувачів.

Метод об'єктно-орієнтованого моделювання передбачає послідовне виконання двох етапів:

- об'єктно-орієнтований аналіз;
- об'єктно-орієнтоване проектування.

Тому термін «об'єктно-орієнтоване моделювання» еквівалентний терміну «об'єктно-орієнтований аналіз та проектування»

Аналіз – широке поняття, його зміст детальніше відображають терміни *аналіз системних вимог* (дослідження вимог до майбутньої програмної системи) та *об'єктний аналіз* (дослідження об'єктів предметної області).

Під *предметною областю* розуміють ту частину реального світу, що має істотне значення чи безпосереднє відношення до процесу функціонування програми. Іншими словами: предметна область містить у собі тільки ті об'єкти і взаємозв'язки між ними, які необхідні для опису вимог і умов розв'язання конкретної задачі.

У загальному випадку виділення базових об'єктів (або компонентів) предметної області є нетривіальною задачею. Складність виявляється у неформальному характері процедур чи правил, які можна застосовувати з цією метою. Окрім того, таку роботу необхідно виконувати спільно з фахівцями чи експертами, що добре знають предметну область.

У процесі *проектування* головну увагу звертають на концептуальні рішення, які забезпечують виконання системних вимог, а не на питання реалізації. У процесі об'єктно-орієнтованого проектування визначають програмні об'єкти та способи їхньої взаємодії або схеми баз даних.

Під час *аналізу системних вимог* необхідно з'ясувати потреби замовника, аналізуючи отриману інформацію від керівництва компанії та майбутніх користувачів системи.

Під час аналізу необхідно визначити:

- функціональні вимоги до системи (або бізнес-процеси), тобто встановити варіанти використання програмної системи для реалізацій конкретних функцій чи дій у даній предметній області;
- потоки даних для кожного бізнес-процесу;
- границі системи;
- користувачів системи та процеси їхньої взаємодії з системою.

У результаті аналізу, зазвичай, оформлюють словник (глосарій) предметної області і технічне завдання, у якому сформульовано функціональні та нефункціональні вимоги до системи.

До нефункціональних вимог відносять питання надійності, зручності використання, продуктивності, можливості супроводу програм, питання безпеки, проектні та апаратні обмеження.

Технічне завдання є гарантією єдиного трактування вимог замовниками і проектувальниками, воно дає змогу вирішувати спірні питання з приводу функцій системи, що виникають у процесі її створення.

В UML певним синонімом терміну «технічне завдання» є специфікація вимог до системи (Software Requirement Specification – SRS), в яких визначаються межі системи, користувачів і функціональні вимоги.

SRS є текстовою основою формалізації етапу постановки задачі за допомогою діаграм прецедентів.

Неможливо визнати один шаблон моделювання найкращим у всіх областях практичного застосування, однак, об'єктно-орієнтоване моделювання найприйнятніше для найширшого кола задач.

Об'єктно-орієнтована модель має чотири головні властивості:

- абстрагування – відокремлення істотних характеристик об'єкта, що відрізняють його з-поміж інших видів об'єктів;
- інкапсуляція – приховування внутрішньої реалізації об'єкта за наданим цим об'єктом інтерфейсом;
- модульність – здатність системи поділятися на зв'язані між собою модулі;
- ієрархія – упорядкування абстракцій та розташування їх за рівнями.

Ці властивості є головними, за відсутності будь-якої з них модель не буде об'єктно-орієнтованою.

Абстрактний граф моделі, який складається з множини різнотипних сутностей і відношень, не підлягає конструюванню загалом. З нього відокремлюють тільки сутності та відношення, що актуальні для певної точки зору (представлення). Набір представлень моделі є формальним та містить:

- представлення прецедентів;
- логічне представлення;
- представлення процесів;
- представлення компонентів;
- представлення розміщення.

Для моделювання програмних систем необхідно розглядати їх з різних позицій. Усі, хто має відношення до проекту, – кінцеві користувачі, аналітики, прикладні програмісти, системні адміністратори, тестувальники, менеджери проектів – переслідують власні інтереси, і кожен дивиться на створену систему по-різному в різні моменти її життя.

2.3 Застосування інструментальних програмних засобів для моделювання об'єктів та процесів геоінформаційних систем

План

1. Використання інструментальних засобів для опису функціональних моделей об'єктів та процесів.
2. Використання інструментальних засобів для моделювання потоків даних.
3. Використання інструментальних засобів для опису логіки взаємодії компонентів об'єктів та процесів.
4. Інструментальні засоби для опису процедур автоматизованої побудови мережевих моделей геоінформаційних систем.

Ознайомимося з інструментальним засобом для опису процедур автоматизованої побудови мережевих моделей ГІС – FPN (Fuzzy Petri Nets), нечіткі мережі Петрі, екран запуску якого подано на рисунку 2.8 [1].

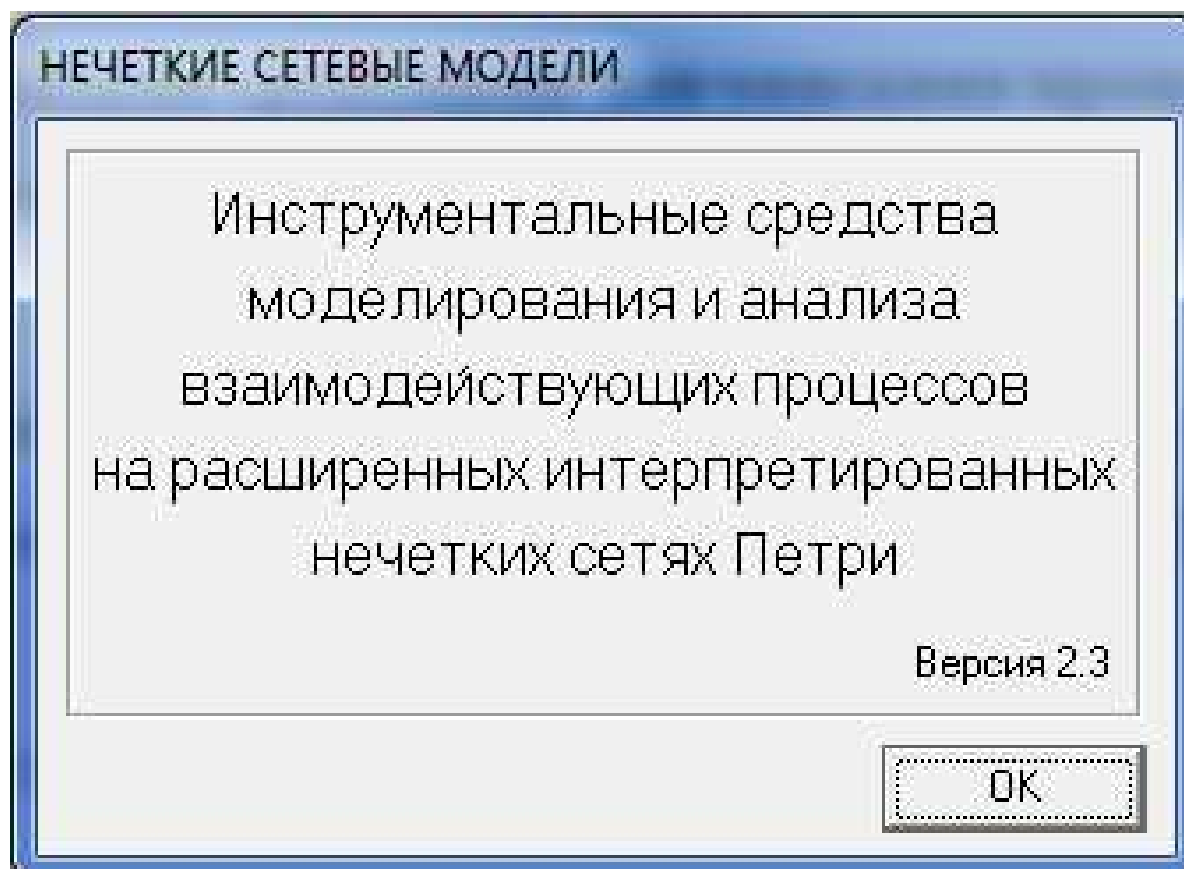


Рисунок 2.8 – Інструментальний засіб для опису процедур
автоматизованої побудови мережевих моделей FPN

У даному інструментальному засобі для опису процедур автоматизованої побудови мережевих моделей ГІС надається ряд можливостей (рис. 2.9) [1]:

- побудова та аналіз;
- моделювання та аналіз;
- навчання;
- побудова;
- моделювання;
- аналіз.

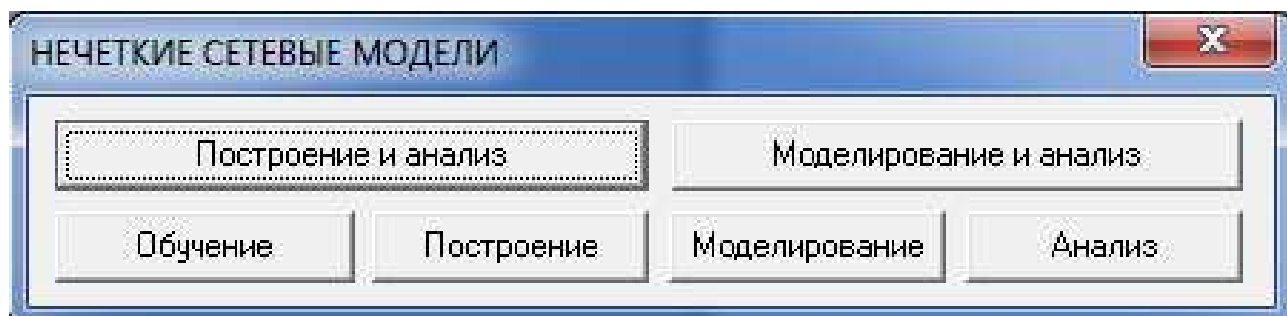


Рисунок 2.9 – Можливості інструментального засобу для опису процедур автоматизованої побудови мережевих моделей FPN

Важливою ланкою побудови сучасних систем виробничого призначення є створення ефективних моделей і засобів їх реалізації. Існує велика кількість моделей, які можуть бути використані в системах моделювання реальних процесів. Перспективним напрямом створення таких засобів є мережі Петрі. Існуючі моделі мають меншу або рівну мережам Петрі потужність моделювання. Мережі Петрі мають ряд суттєвих переваг, що робить їх перспективним апаратом моделювання складних процесів серед інших.

На ринку програмних продуктів існує ряд програм, що певною мірою дозволяють автоматизувати процеси, пов'язані з побудовою складних структур даних. Серед них слід зазначити Visual Case 6.0 фірми Rogue Wave Software, яка інтегрується в MS Visual Studio і є засобом візуального проектування структури програмного засобу.

В основу функціонування цього засобу покладено можливість створення і перегляду існуючих зв'язків між окремими структурними елементами програми. Застосування цього продукту для моделювання програм викликає труднощі, програма забезпечує тільки візуалізацію класів, але не забезпечує їх моделювання та обробку.

Для моделювання складних процесів виникає потреба в автоматизованій процедурі побудови мереж Петрі. Складність процедури візуального проектування мережі Петрі полягає в динамічності структури даних, що відображає компоненти мереж Петрі, а також у необхідності вирішення проблеми автоматичного розподілу пам'яті під кожен елемент уявлення мережі.

Створення візуального інтерфейсу побудови мереж Петрі надає можливість моделювати більш складні процеси. Процедура побудови мережі Петрі була реалізована на мові MS Visual C++ 6.0 у вигляді програмного засобу PK_NET Graph Edition.

Основна частина програми, що реалізує саме подання мережі Петрі складається з шести класів [1]:

- CGENormalShema (блок-схема);
- CGELink (зв'язок);
- CGECtrlPoint (точка зв'язку);
- CGEPetriShema (вершина мережі Петрі);
- SpetriShemaLink (зв'язок у мережі Петрі);
- CHMatrix (матриця інцидентності).

Блок-схема складається з подання (CGENormalShema) і точок управління зв'язками (CGECtrlPoint). Кожна блок-схема має своє подання у вигляді вершин і їх інцидентності в мережі Петрі.

Для забезпечення контрольованості подання блок-схеми в мережі Петрі, клас CGENormalShema зберігає адреси відповідних їй вершин мережі Петрі. Вершина мережі Петрі статична і не дозволяє маніпуляцій зі зв'язками, тому не має в своєму складі контрольних точок.

Взаємодія окремих об'єктів між собою здійснюється за рахунок динамічного визначення предків класу та дозволяє редагувати модель. За рахунок цього організована інтерактивна операція зв'язку двох схем між собою через об'єкт зв'язку. Об'єкт-зв'язок є автономним об'єктом схеми, введений в структуру для можливості розташування точок зв'язку так, щоб він мав наочний вигляд.

Програма надає можливість автоматизувати процес побудови мережі Петрі. Програма оснащена «дружнім інтерфейсом», який надає можливість користуватися програмним засобом користувачеві не програмісту.

Кожна блок-схема виведена на панель інструментів і є окремим елементом візуального відображення моделі. Для перегляду моделі у вигляді матриці інцидентності введено спеціальне подання у вигляді таблиці, що автоматично заповнюється при побудові моделі.

Програма забезпечує інтерактивну побудову моделі за допомогою звичайних блок-схем, їх вільне переміщення на віконному поданні.

Ефективність розробки підтверджена моделюванням реальних процесів.

Програмою запропоновано (рис. 2.10) [1]:

- панель опцій (1), що надає можливість управляти загальними можливостями програми;
- меню програми (2), що дублює кнопки швидкого доступу;
- панель конструктора моделі (3), за допомогою цієї панелі створюються моделі, що підлягають перетворенню в мережу Петрі;
- віконне подання матриці інцидентності (4);
- подання області редагування моделі за допомогою блок-схем (5);
- подання області перегляду моделі у вигляді мережі Петрі (6);
- рядок стану програми (7).

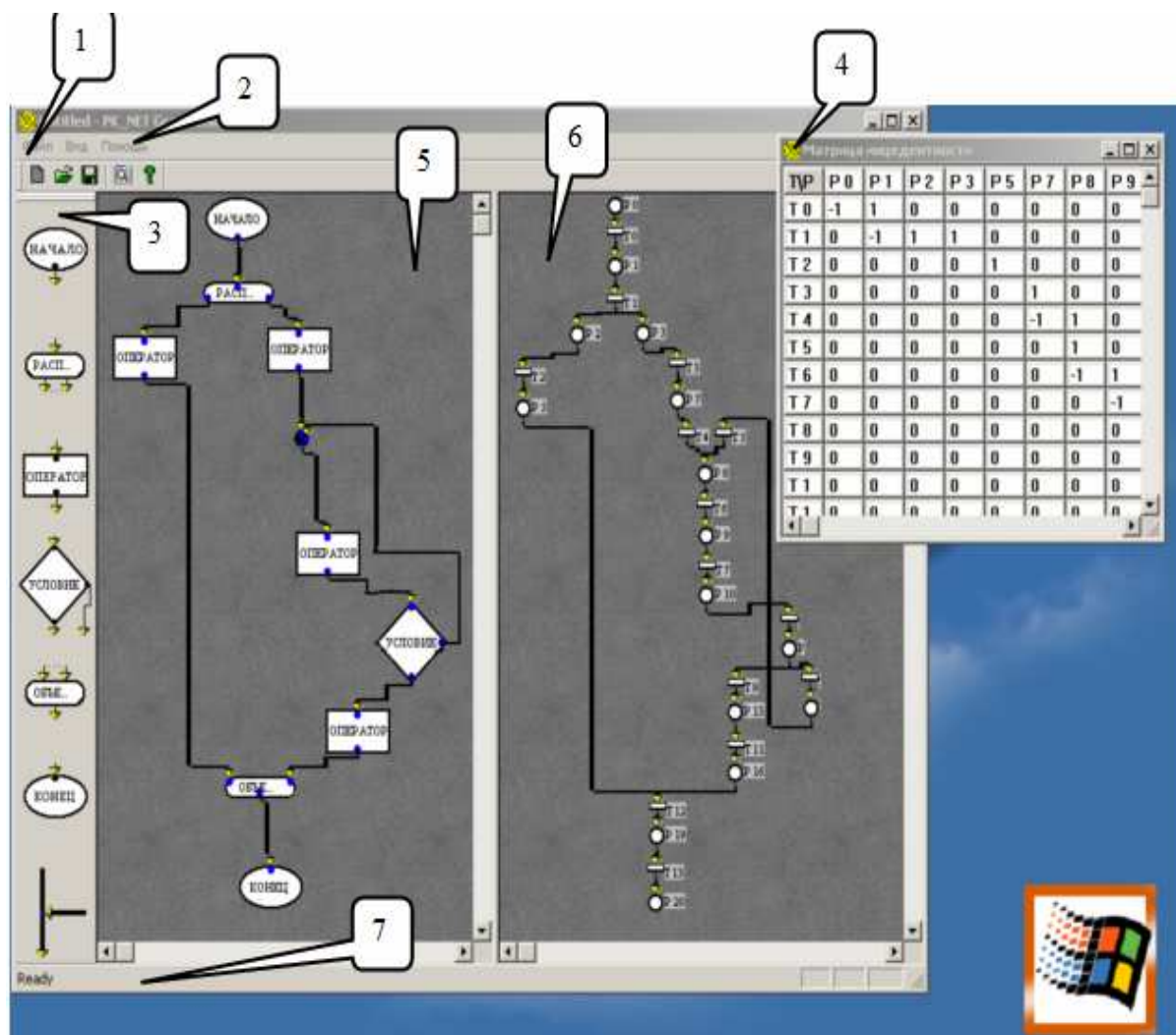


Рисунок 2.10 – Можливості інструментального засобу для опису процедур автоматизованої побудови мережевих моделей FPN

Створення моделі здійснюється за допомогою панелі конструктора.

Щоб створити блок-схему, на панелі блок-схем слід натиснути ліву кнопку «миші», далі перемістити курсор «миші» у потрібне місце і знову натиснути ліву кнопку «миші».

Інтерпретація мережевих моделей подана на рисунку 2.11 [1].

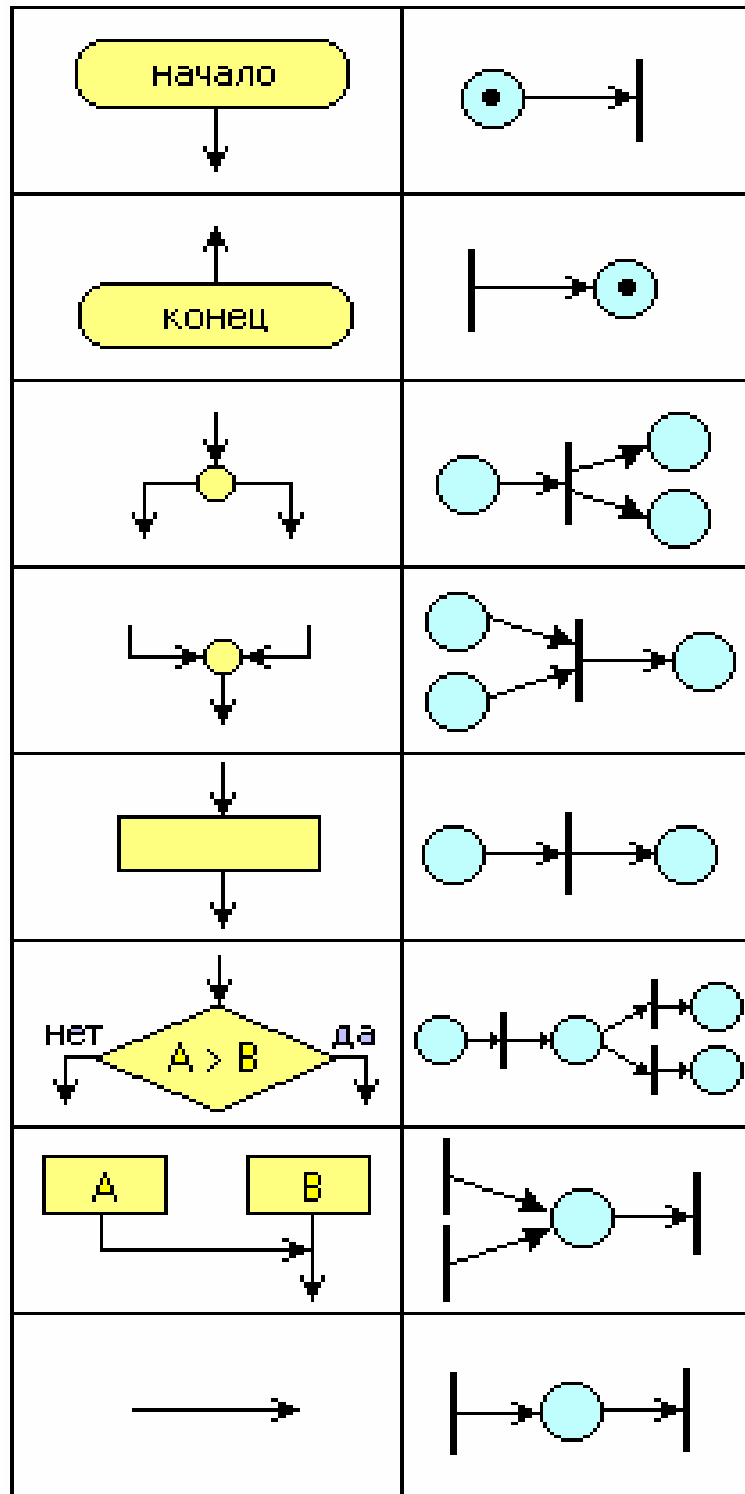


Рисунок 2.11 – Інтерпретація мережевих моделей

Для переміщення елементів моделі слід натиснути ліву кнопку «миші» і, не відпускаючи її, перемістити курсор «миші» в потрібне місце. Щоб зафіксувати положення елемента моделі необхідно відпустити кнопку «миші».

Щоб створити логічний зв'язок між двома елементами моделі слід навести курсор «миші» на блакитну крапку, зображену на блок-схемі, натиснути ліву кнопку «миші» і утримуючи її навести курсор «миші» на точку іншої блок-схеми та відпустити кнопку «миші». Після цього буде створено зв'язок двох схем, а на МП буде створено зв'язок двох вершин.

Для вилучення блок-схеми з загального подання моделі слід натиснути праву кнопку «миші» над потрібної блок-схемою і з меню вибрати пункт «Видалити», після цього блок-схема і її подання на МП буде видалено з моделі.

Для вилучення елементу зв'язку між двома блок-схемами слід навести курсор «миші» на потрібний зв'язок і, натиснувши праву кнопку «миші», вибрати пункт меню «Видалити».

Зміна зв'язків забезпечується за рахунок вилучення зв'язку з моделі і відповідно створення нового зв'язку з іншої блок-схемою. У процесі зміни зв'язків змінюється матриця інцидентності та подання МП.

Щоб отримати інтерактивну підказку слід навести курсор «миші» на потрібний об'єкт програми, через 1-2 секунди на екрані буде відображена підказки для цього об'єкта.

Користувач має можливість зберегти інформацію щодо моделі у вигляді файлів *.Mtr – матриці інцидентності і *.Nmr – вектора початкової маркірування за допомогою меню «Файл – Зберегти як».

Приклад моделі подано на рисунку 2.12, алгоритм якої має такий вигляд:

Крок 1. Початок.

Крок 2. Реалізація процесу А.

Крок 3. Виконання умови: якщо виконується умова $A > 0$, то переходимо до кроку 4, інакше – до кроку 5.

Крок 4. Реалізація процесу В.

Крок 5. Реалізація процесу С.

Крок 6. Виконання розгалуження: реалізація процесу Д через розбиття на підпроцес Е та підпроцес Ф.

Крок 7. Реалізація підпроцесу Е.

Крок 8. Реалізація підпроцесу Ф.

Крок 9. Виконання злиття: реалізація процесу Г на основі результатів підпроцесу Е та підпроцесу Ф.

Крок 10. Кінець.

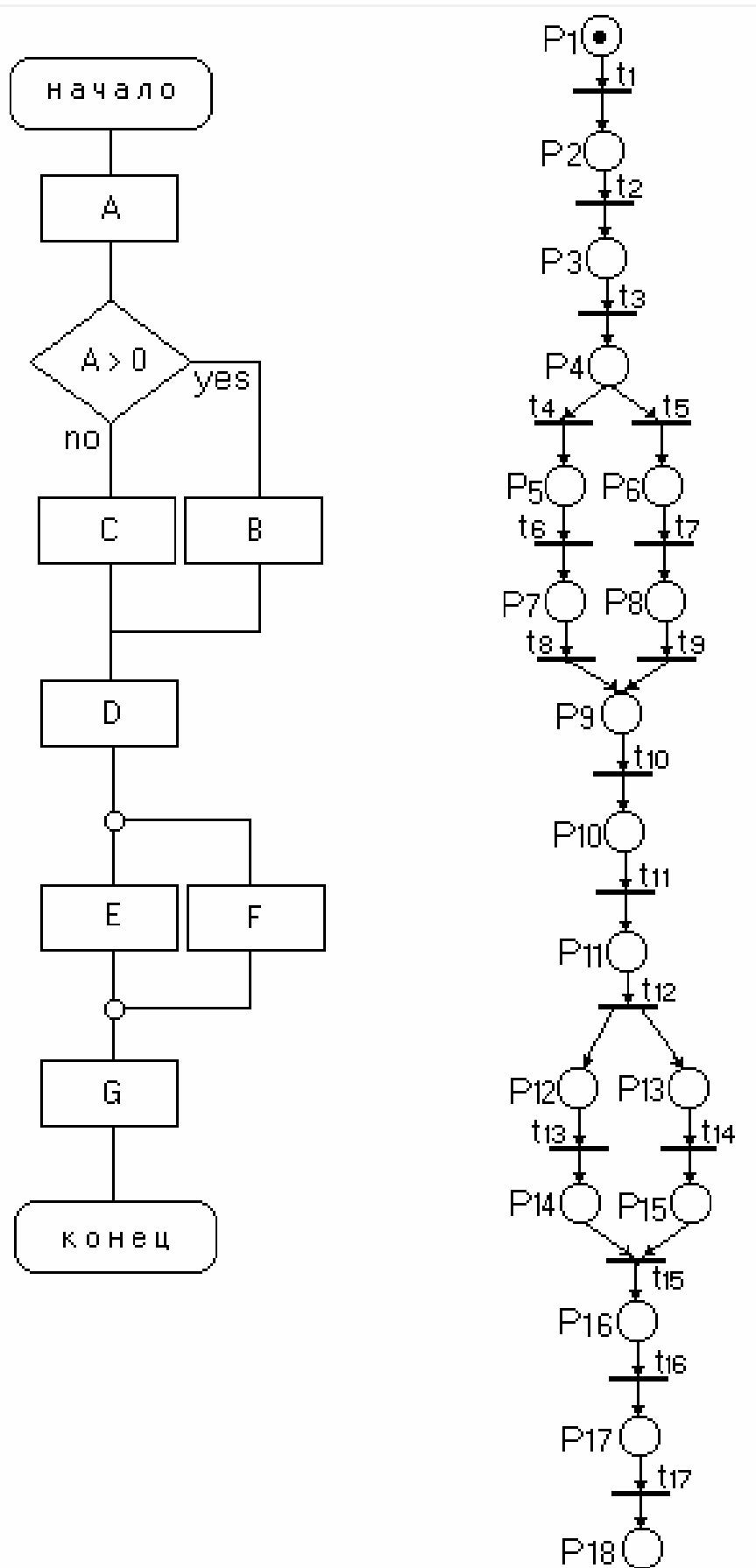


Рисунок 2.12 – Приклад моделі

Сучасний процес математичного моделювання важко реалізувати без комп'ютерів або вбудованих контролерів. Для цього використовуються різноманітні інструментальні програмні засоби та середовища (MathCad, MatLab, Mathematica, Maple, Derive, VisSim, Genius), що суттєво спрощують моделювання.

Сучасні математичні пакети можна використовувати як звичайний калькулятор, як засоби для спрощення виразів при розв'язанні будь-яких математичних задач, як генератори графіки або навіть звука.

Звичайно, кваліфікований користувач, який в достатній мірі володіє однією з мов програмування (C++, Java, Pascal, Fortran, Prolog), може самостійно створити окрему програму або комплекс програм, що дозволить реалізувати алгоритм його задачі. Такий підхід потребує, як правило, великих працевитрат на програмування, для скорочення часу програмування було створено прикладні програмні пакети, області використання яких у значній мірі перекриваються.

Для найбільш ефективного використання обчислювальної техніки необхідно правильно вибрати найкращий пакет програм на ранній стадії розв'язання прикладної задачі.

Під час дослідження систем автоматичного регулювання, обчислювальних математичних задач, найбільш ефективним є використання програмної системи Matlab з широким класом предметно-орієнтованих бібліотек (toolbox) та інструментом візуального моделювання Simulink.

У системі MatLab існують широкі можливості для програмування, її бібліотека C Math є об'єктною і містить понад 300 процедур обробки даних мовою C. У середині пакета можна використовувати як процедури самої MatLab, так і стандартні процедури мови C, що робить цей інструмент підмогою при розробці додатків.

Для візуалізації моделювання система MatLab має бібліотеку Image Processing Toolbox, що забезпечує широкий спектр функцій, що підтримують візуалізацію проведених обчислень безпосередньо із середовища MatLab, аналіз, а також можливість побудови алгоритмів обробки зображень.

Серед інших бібліотек системи MatLab можна також відзначити System Identification Toolbox – набір інструментів для створення математичних моделей динамічних систем, заснованих на вхідні та вихідні дані. Особливістю цього інструменту є наявність гнучкого користувацького інтерфейсу, що дозволяє організувати дані й моделі. Бібліотека System Identification Toolbox підтримує як параметричні, так і непараметричні методи.

Інтерфейс системи полегшує попередню обробку даних, роботу з ітеративним процесом створення моделей для одержання оцінок і виділення найбільш значимих даних. Що стосується математичних обчислень, то MatLab надає доступ до великої кількості підпрограм, що містяться в бібліотеці NAG Foundation Library компанії Numerical Algorithms Group Ltd. Це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики. Сьогодні система MatLab широко використовується в техніці, науці та освіті, але все-таки вона більше підходить для аналізу даних і організації обчислень.

Для візуального моделювання та моделювання сумісно з реальною апаратурою більш зручним є програмний пакет VisSim.

Аналітичні перетворення дозволяють виконувати більшість математичних програмних продуктів MathCad, Mathematica, Maple. З цих трьох поширених математичних пакетів, найбільш потужним є Maple. Maple надає зручне середовище для комп'ютерних експериментів. Пакет дозволяє створювати інтегровані середовища за участю інших систем і універсальних мов програмування високого рівня. Коли розрахунки зроблені й потрібно оформити результати, то можна використовувати засоби цього пакета для візуалізації даних і підготовки документації для публікації. Робота проходить інтерактивно – користувач вводить команди і відразу бачить на екрані результат їхнього виконання. Ядро символьних обчислень Maple включено до складу цілого ряду систем комп'ютерної математики – від систем для широкого кола користувачів типу MathCad до однієї із кращих систем для чисельних розрахунків і моделювання MatLab.

Імітаційне моделювання – це метод дослідження, при якому досліджувана система замінюється моделлю, що з достатньою точністю описує реальну систему, з нею проводяться експерименти з метою одержання інформації про цю систему.

Під *імітаційною системою* розуміють програмний або апаратно-програмний комплекс, призначений для вирішення завдань із використанням методу імітаційного моделювання. При виділенні різновидів імітаційних систем виходять із того, що вони є інструментальними засобами, що забезпечують автоматизовану підтримку певних видів діяльності користувача.

Імітаційна система реалізує алгоритм вирішення завдання і надає користувачеві сервісні можливості по управлінню обчислювальним процесом. Автоматизована підтримка інших етапів системного аналізу засобами імітаційної системи не є обов'язковою. Однак, саме ступінь автоматизації визначає можливості імітаційної системи і є основою класифікації.

З урахуванням етапності системного аналізу і технологічних завдань, що розв'язуються на них, виділимо можливий набір засобів імітаційної системи, що автоматизують виконання ряду функцій, реалізованих на цих етапах:

– створення моделі може бути підтримано такими засобами автоматизації:

- 1) частково готовою моделлю або моделями;
- 2) компіляторами з алгоритмічної мови високого рівня, спеціалізація якого полегшує процес складання алгоритмів імітації;
- 3) спеціальною мовою високого рівня, що дозволяє виконати інформаційний або математичний опис моделі системи;
- 4) конверторами моделей, що дозволяють здійснювати перетворення моделей одного виду в моделі іншого виду (інформаційної в математичну, математичної в імітаційну, інформаційної в імітаційну);
- 5) засобами контролю погодженості різних видів моделей з концептуальним поданням моделі;

– перевірка адекватності та технічної реалізації може виконуватися з використанням:

- 1) програм обчислення показників адекватності;
- 2) автоматизованої технології проведення обмеженого експерименту з імітаційною моделлю;
- 3) програм обчислення характеристик складності моделі;
- 4) програм обчислення ресурсних показників методу вирішення завдання;

– корекція моделі може забезпечуватися:

- 1) автоматизованими технологіями редагування текстів моделей;
- 2) програмами еквівалентних перетворень математичних і алгоритмічних моделей заданого класу;

– створення алгоритму вирішення завдання може підтримуватися:

- 1) методо-орієнтованими бібліотеками та пакетами програм;
- 2) конструкторами алгоритмів вирішення завдань;
- 3) інформаційними системами підтримки прийняття рішень;

– складання і уточнення схеми вирішення завдання може виконуватися з використанням:

- 1) програмних засобів контролю інформаційної сумісності сімейства завдань;
- 2) редакторів текстів обчислювальних схем.

При виборі засобів імітаційного моделювання варто враховувати всі можливості, що вони надають, які можна об'єднати у такі групи:

- основні характеристики;
- сумісне програмне забезпечення;
- анімація;
- статистичні можливості;
- звіти з вихідними даними і графіками;
- послуги, що надаються замовникам і документація.

Найбільш популярними пакетами імітаційного моделювання є:

- Arena компанії Rockwell Automation;
- AnyLogic компанії XJ Technologies;
- GPSS World фірми Minuteman Software;
- Process Charter 1.0.2 компанії Scitor;
- Powersim 2.01 фірми Modell Data AS;
- Ithink 3.0.61 виробництва High Performance Systems;
- Extend+BPR 3.1 компанії Imagine That!;
- Vensim фірми Ventana Systems.

Ці пакети найбільше відрізняються стилем моделювання, тобто середовищем, за допомогою якого створюються моделі. У пакеті Process Charter модель будується за допомогою блок-схеми. Powersim і Ithink використовують систему позначень Systems Dynamics, запропоновану в 1961 році Джейм Форрестером Массачусетського технологічного інституту. Extend застосовує компоновочні блоки. Всі продукти, крім Process Charter, дозволяють проводити аналіз чутливості, тобто багаторазово виконувати експерименти на моделі із різними вхідними параметрами, щоб зрівняти результати декількох прогонів.

Розглянемо найбільш поширені пакети для імітаційного моделювання [2].

GPSS World (GPSSW, General Purpose System Simulation World – Світова загальноцільова система моделювання), розроблена для ОС Windows. Цей програмний продукт увібрав у себе весь арсенал новітніх інформаційних технологій. Він включає розвинені графічні оболонки для створення моделей і інтерпретації вихідних результатів моделювання, засоби мультимедіа та відео, об'єктно-орієнтоване програмування. В основу системи GPSS World покладена мова імітаційного моделювання GPSS (General Purpose System Simulation – Загальноцільова система моделювання).

Система GPSS World – потужне універсальне середовище моделювання як дискретних, так і безперервних процесів, призначене для професійного моделювання найрізноманітніших процесів і систем.

За допомогою цієї системи, наприклад, можна ефективно моделювати як виробничі, так і невиробничі процеси: функціонування торговельних і розважальних закладів, портів, вуличного руху, проведення воєнних дій, роботу редакцій, установ і мережі інтернет, різних систем масового обслуговування. Система має великий набір команд для управління процесом моделювання, які можна як використовувати в інтерактивному режимі, так і включати в модель. Забезпечено можливість проведення експериментів, згенерованих системою, користувацьких і оптимізаційних. У системі GPSSW реалізована процедура візуалізації процесу функціонування моделі з використанням методів мультиплікації, для швидкого виправлення помилок використовується текстовий редактор.

Система GPSSW досить проста у вивченні і універсальна в застосуванні. Ефективне використання системи передбачає виконання ряду етапів:

- постановка задачі;
- виявлення основних особливостей;
- створення імітаційної моделі процесу;
- подання імітаційної моделі в системі GPSSW;
- моделювання системи.

Пакет імітаційного моделювання **Rockwell Arena**.

Система Arena компанії Rockwell Automation є поширеною на ринку програм імітаційного моделювання. В Arena використовується процесор і мова імітаційного моделювання SIMAN. Основні області застосування даного пакета: виробництво (моделювання конвеєрного виробництва, визначення вузьких місць), логістика й складське господарство (оптимізація використання складських площ), озброєння і безпека, медицина (моделювання потоку пацієнтів, розподіл персоналу).

Arena надає користувачеві зручний графічний інтерфейс із набором шаблонів моделюючих конструкцій. Для створення моделі в пакеті Arena моделюючі конструкції спочатку перетягують у вікно моделі, а потім з'єднують, щоб позначити рух об'єктів у системі. Потім моделюючі конструкції деталізуються за допомогою діалогових вікон або вбудованих таблиць.

В ієрархії моделі може бути необмежене число рівнів. Rockwell Arena випускається тільки для операційної системи Windows. В Arena передбачений експорт даних з Microsoft Excel і Microsoft Access.

Число потоків випадкових чисел у пакеті Arena не обмежено, користувач має доступ до 12 стандартних теоретичних розподілів ймовірностей, а також до емпіричних розподілів.

Arena забезпечує виведення на екран двомірної та тривимірної (Arena 3DPlayer) анімації, дозволяє виводити на екран динамічну графіку (гістограми та графіки тимчасової залежності).

Пакет імітаційного моделювання **AnyLogic**.

AnyLogic – програмне забезпечення для імітаційного моделювання складних систем і процесів, розроблене російською компанією XJ Technologies.

Графічне середовище AnyLogic побудоване по тому ж принципі, що і у Rockwell Arena, моделюючі конструкції розташовуються в палітрах (аналог шаблонів в Arena). Для створення моделі, як і в Arena, моделюючі конструкції перетягують в область моделі і з'єднують. Деталізувати моделюючі конструкції можна, виділивши їх і змінивши параметри, використовуючи панель властивостей. AnyLogic підтримує ієрархічне моделювання, а також створення власних моделюючих конструкцій і об'єднання їх у бібліотеки (тільки для версії Professional). AnyLogic заснований на Java і базується на платформі Eclipse – сучасному стандарті для бізнесів-додатків. Завдяки Eclipse AnyLogic працює на всіх поширених операційних системах (Windows, Mac, Linux).

У редакторі AnyLogic можливо розробити анімацію та інтерактивний графічний інтерфейс моделі. Анімація може бути ієрархічною і підтримувати кілька перспектив. В AnyLogic користувачеві доступно 29 стандартних теоретичних розподілів, є можливість зафіксувати набір випадкових чисел і зробити абсолютно ідентичний експеримент.

Для створення звітів в AnyLogic відведена спеціальна палітра «Статистика», у якій утримуються конструкції для збору даних щодо ходу роботи моделі, а також знаходяться різні діаграми, графіки та гістограми.

Система **BPsim** спирається на апарат динамічних експертних систем і значною мірою вільна від типових недоліків, у ній визначені нові класи об'єктів: операції, ресурси, засоби, процеси, джерела, споживачі ресурсів, перехрестя, параметри. Опис причинно-наслідкових зв'язків задається спеціальними об'єктами.

Система автоматизованого проектування **Modelling** є засобом візуального проектування дискретних і неперервно-дискретних імітаційних моделей, а також є набором компонентів та шаблонів проектів для середовища програмування Borland Delphi. Головна сфера застосування системи імітаційного моделювання Modelling – розробка імітаційних моделей і спеціалізованих систем імітаційного моделювання під час вивчення і проектування складних комплексних систем, де використання стандартних засобів проектування (Mathlab, GPSS) незручне або неефективне.

ProcessModel дозволяє створювати ієрархічні моделі для кращої організації та управління великими модельними проектами. Групи розробників можуть створювати різні частини складної моделі, а потім, об'єднавши їх, моделювати процес в цілому.

Крім розглянутих спеціалізованих пакетів для імітаційного моделювання, пакет розширення **MatLab Simulink for Windows** також служить для імітаційного моделювання моделей, що складаються із графічних блоків із заданими властивостями (параметрами). Компоненти моделей, у свою чергу, є графічними блоками і моделями, які містяться в ряді бібліотек і за допомогою миші можуть переноситися в основне вікно та з'єднуватися один з одним необхідними зв'язками.

До складу моделей можуть включатися джерела сигналів різного виду, віртуальні прилади, що реєструють, графічні засоби анімації. Подвійне клацання мишею на блоці моделі виводить вікно зі списком його параметрів, які користувач може змінювати. Запуск імітації забезпечує математичне моделювання побудованої моделі з наочним візуальним поданням результатів.

Пакет заснований на побудові блокових схем шляхом переносу блоків з бібліотеки компонентів у вікно редагування створеної користувачем моделі. Потім модель запускається на виконання. Можливе моделювання складних систем, що складаються з множини підсистем.

Метод імітаційного моделювання покладений в основу нового сімейства *пакетів підтримки прийняття рішень* на основі побудови поточних моделей замкнених систем. Він добре зарекомендував себе для розв'язування задач фундаментального аналізу і стратегічного планування.

Розглянемо ряд пакетів, що побудовані з використанням цього методу.

Пакет **Ithink Analyst 5.0** (HPS) – це компактний, об'єктно-орієнтований пакет прикладних програм з Desktop-інтерфейсом, що забезпечує графічну, обчислювальну та інформаційну підтримку процедур високорівневого системного аналізу складних процесів організації управління.

У прикладному пакеті Ithink знайшли своє вдале втілення ідеї структурного проектування Е. Йордана, структурного аналізу Т. Демарко, системного аналізу С. Гейна і Т. Сарсона.

Використання Ithink забезпечує, насамперед, можливість розробки моделей різноманітних систем. Особливо важко описувати системи «середньої» складності, труднощі опису таких систем виникають із-за великої кількості компонентів і взаємозв'язків.

Ithink дає змогу розв'язувати традиційні для системного аналізу завдання:

- побудова моделей систем «середньої» складності як сукупності простих;

- визначення механізму взаємодії компонентів системи з метою поведінкового опису в термінах її основних характеристик;

- використання і розробка формальних ефективних процедур імітації відповідних поведінкових описів для одержання статистичних оцінок і визначення якісних характеристик;

- формування процедур генерації представницьких тестових наборів, необхідних для дослідження нетривіальної поведінки;

- цілеспрямоване вивчення поведінки моделей для складання попередніх розкладів роботи з компонентами системи.

Ithink можна застосовувати як потужний корпоративний засіб етапу концептуального аналізу великих проєктів. Практика свідчить, що замовник практично ніколи не в змозі адекватно оцінити складний проєкт. Особливий інтерес становлять методи і засоби, що дають змогу з мінімальними витратами здійснити опис функціональної архітектури макету майбутньої системи.

Програма **Powersim Enterprise Kit** – це програма системного моделювання, вона призначена для тих, хто вирішив застосувати можливості системної динаміки для розв'язання завдань.

Методи імітаційного моделювання досить успішно застосовуються під час аналізу процесів дорожнього руху, проходження тунелів, черг біля світлофорів, технічного огляду автомашин, транспортних потоків на автомагістралях, вулицях міст, перехрестях, на окремих ділянках руху [9, 10].

Під час моделювання роботи залізничного транспорту можна визначити місце розташування додаткових колій, їх необхідну довжину та місця встановлення додаткових стрілкових переводів, а також розрахувати необхідну кількість вагонів і кількість локомотивів [9, 10].

Використавши імітаційні моделі можна промодельовати повітряний простір у зоні аеропорту, виявити ймовірності конфліктних ситуацій, а також промодельовати рух літаків в аеропорту, проаналізувати процеси реєстрації пасажирів, диспетчерської служби аеропорту, резервування квитків [9, 10].

Імітаційні моделі можна застосувати для моделювання та оцінки якості обслуговування об'єктів житлово-комунального господарства (ремонт будівель, прибирання території, обслуговування ліфтів, ремонт тепломереж, мереж водопостачання та водовідведення, вивезення побутового сміття) [9, 10].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи, моделі та інформаційні технології оцінювання станів складних об'єктів: монографія / Є. І. Кучеренко, В. Є. Кучеренко, І. С. Глушенкова, І. С. Творошенко; ХНАМГ, ХНУРЕ. – Х. : ХНАМГ; ХНУРЕ, 2012. – 278 с.
2. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Издательский дом «Вильямс», 2003. – 864 с.
3. Методичні вказівки до виконання лабораторних, розрахунково-графічних та самостійних робіт з дисципліни «Основи моделювання складних систем» (для студентів 2 курсу денної та заочної форм навчання напряму підготовки 6.080101 – Геодезія, картографія та землеустрій) / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова; уклад.: І. С. Творошенко. – Харків : ХНУМГ ім. О. М. Бекетова, 2015. – 99 с.
4. Кучеренко Є. І. Сіткові моделі в задачах аналізу складних систем: навч. посібник для вузів / Є. І. Кучеренко. – Х. : ХТУРЕ, 1999. – 99 с.
5. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.
6. Томашевський В. М. Моделювання систем / В. М. Томашевський. – К. : Видавнича група ВНУ, 2005. – 352 с.
7. Советов Б. Я. Моделирование систем / Б. Я. Советов, С. А. Яковлев. – М. : Высш. Школа, 2001. – 343 с.
8. Кучеренко Е. И. Прикладные аспекты моделирования нечетких процессов в сложных системах / Е. И. Кучеренко, И. С. Творошенко // Сборник научных трудов ХУВС. – 2010. – Вып. 1 (23). – С. 127-131.
9. Творошенко І. С. Конспект лекцій з дисципліни «Геоінформаційні системи в управлінні територіями» (для студентів 5 курсу денної форми навчання спеціальностей 7.08010105 – Геоінформаційні системи та технології, 8.08010105 – Геоінформаційні системи та технології та студентів 6 курсу заочної форми навчання спеціальності 7.08010105 – Геоінформаційні системи та технології) / І. С. Творошенко; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2015. – 87 с.
10. Баркалов С. А. Модели и механизмы в управлении организационными системами / С. А. Баркалов, В. Н. Бурков, Д. А. Новиков, Н. А. Шульженко. – М. : Тульский полиграфист, 2003. – Том 1. – 560 с.

Навчальне видання

ТВОРОШЕНКО Ірина Сергіївна

КОНСПЕКТ ЛЕКЦІЙ

з дисципліни

**ОСНОВИ МОДЕЛЮВАННЯ
СКЛАДНИХ СИСТЕМ**

*(для студентів 2 курсу заочної форми навчання напряму підготовки
6.080101 – Геодезія, картографія та землеустрій)*

Відповідальний за випуск: *К. А. Мамонов*

За авторською редакцією

Комп'ютерний набір: *І. С. Творошенко*

Комп'ютерне верстання: *І. В. Волосожарова*

План 2015, поз. 17 Л

Підп. до друку 09.11.2015 р.

Друк на різнографі

Тираж 50 пр.

Формат 60x84/16

Ум. друк. арк. 4,0

Зам. №

Видавець і виготовлювач:

Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Революції, 12, Харків, 61002

Електронна адреса: rectorat@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 4705 від 28.03.2014 р.